Smile
OPEN SOURCE SOLUTIONS

Livre blanc

Gestion de contenus

les solutions open source

Patrice Bertrand – Directeur Général
Badr Chentouf – Consultant

Version 3
Pour plus d'information : www.smile.fr
Tél : 01 41 40 11 00
Mailto : sdc@smile.fr

PREAMBULE

Smile

Fondée en 1991, Smile est une société d'ingénieurs experts dans la mise en œuvre de solutions Internet et intranet. Le métier de Smile couvre trois grands domaines :

- La conception et la réalisation de sites Internet haut de gamme. Smile a construit quelques uns des plus grands sites du paysage web français, avec des références telles que Cadremploi ou Explorimmo.
- Les applicatifs Intranet, qui utilisent les technologies du web pour répondre à des besoins métier. Ces applications s'appuient sur des bases de données de grande dimension, et incluent plusieurs centaines de pages de transactions. Elles requièrent une approche très industrielle du développement.
- La gestion de contenu et les outils open source, avec plus de 250 projets de sites Internet, Intranet ou systèmes de gestion documentaire.

www.smile.fr

Ce livre blanc

Ce papier n'est pas un simple comparatif entre les CMS open source que nous avons sélectionnés. Nous nous efforçons d'en faire aussi une introduction aux concepts fondamentaux des CMS et de leur mise en œuvre. La gestion de contenus n'est pas qu'une affaire d'outils, et avant de choisir un outil, il convient d'avoir bien analysé les finalités de son projet.

C'est pourquoi nous proposons, pour chacun des concepts les plus importants, notre analyse des enjeux et des possibilités qui différencient les outils du marché.

Après cette analyse, nous essayons de positionner chacun des CMS à l'étude.

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Quelques références de Smile

Intranets - Extranets

- Société Générale - Caisse d'Épargne - Bureau Veritas - Commissariat à l'Energie Atomique
- Visual - Vega Finance - Camif - Lynxial - RATP - AMEC-SPIE - Sonacotra - Faceo - CNRS
- AmecSpie - Château de Versailles - Banque PSA Finance - Groupe Moniteur - CIDJ - CIRAD
- Bureau Veritas - Ministère de l'Environnement - JCDecaux - Ministère du Tourisme
- DIREN PACA - SAS - Institut National de l'Audiovisuel - Cogedim - Ecureuil Gestion
- IRP-Auto - AFNOR - Conseil Régional Ile de France - Verspieren - Zodiac – OSEO - Prolea
- Conseil Général de la Côte d'Or - IPSOS - Bouygues Telecom - Pimkie Diramode
- Prisma Presse - SANEF - INRA – HEC

Internet, Portails et e-Commerce

- cadremploi.fr - chocolat.nestle.fr - creditlyonnais.fr - explorimmo.com - meilleurtaux.com
- cogedim.fr - capem.fr - editions-cigale.com - hotels-exclusive.com - souriau.com - pci.fr
- gdf.fr/presse - dsv-cea.fr - egide.asso.fr - osmoz.com - spie.fr - nec.fr - sogeposte.fr
- metro.fr - stein-heurtey-services.fr - bipm.org - buitoni.fr - aviation-register.com - cci.fr
- schneider-electric.com - calypso.tm.fr - inra.fr - cnil.fr - longchamp.com - aesn.fr
- Dassault Systemes 3ds.com - croix-rouge.fr - worldwatercouncil.org - projectif.fr
- editionsbussiere.com - glamour.com - fratel.org - tiru.fr - faurecia.com - cidil.fr - prolea.fr
- ETS Europe - ecofi.fr - credit-cooperatif.fr - odit-france.fr - pompiersdefrance.org
- watermonitoringaliance.net - bloom.com - meddispar.com - nmmedical.fr - medistore.fr
- Yves Rocher - jcdecaux.com - cg21.fr - Bureau Veritas veristar.com - voyages-sncf.fr
- eurostar.com - AON conseil - OSEO - cea.fr - eaufrance.fr - banquepsafinance.com
- nationalgeographic.fr - idtgv.fr - prismapub.com - Bouygues Construction
- Hachette Filipacchi Media - ELLE.fr - femmeactuelle.fr - AnnoncesJaunes.fr

Applications métier, systèmes documentaires, business intelligence

- Renault - Le Figaro - Sucden - Capri - Libération - Société Générale - Ministère de l'Emploi
- CNOUS - Neopost Industries - ARC - Laboratoires Merck - Egide - Bureau Veritas
- ATEL-Hotels - Exclusive Hotels - Ministère du Tourisme - Groupe Moniteur - Verspieren
- Caisse d'Epargne - AFNOR - Souriau - MTV - Capem - Institut Mutualiste Montsouris
- Dassault Systemes - Gaz de France - CFRT - Zodiac - Croix-Rouge Française
- Centre d'Information de la Jeunesse (CIDJ) - Pierre Audoin Consultants - EDF R&D

Version 2.1

Téléchargée à plus de 10 000 exemplaires, la version 1 de ce livre blanc remontait à 2004, et avait besoin d'une mise à jour. Cette version 2.1 est une refonte d'envergure.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Tout d'abord, **nous avons éliminé de notre panel la solution Redhat CMS**, et ceci pour plusieurs raisons : Redhat est aujourd'hui devenu Byline, sous la responsabilité du consortium ObjectWeb, mais le transfert est loin d'être finalisé et le développement semble arrêté. De plus, ObjectWeb a déjà une solution de portail et de gestion de contenus, avec l'intégration récente de eXoplatform, projet plus dynamique. Il nous paraît difficile dans ce contexte de croire à la pérennité de cette solution, par ailleurs peu répandue.

En revanche, **nous avons intégré Apache Lenya et InfoGlue**, deux produits en environnement Java, répondant à des besoins bien distincts. Ces deux solutions sont de qualité et ont déjà plusieurs références à leur actif, ce qui permet de compléter l'offre open source J2EE, jusqu'ici relativement réduite.

Enfin, nous avons mis à jour nos commentaires avec les évolutions importantes des différentes solutions. **Typo3 a notamment beaucoup évolué**, avec les extensions TemplaVoila pour les structures de contenus, et les extensions de gestion de versions et de workflow qui lui manquaient tant. **OpenCMS a lui aussi profité** de sa dernière version pour se doter de structuration de contenus, tandis que **eZ publish** apporte un lot impressionnant d'amélioration dans ses version **3.6, puis 3.7**.

www.smile.fr

Version 3.0

Après plus de 25 000 téléchargements, voici une nouvelle mise à jour majeure de cet ouvrage. Nous avons souhaité ajouter deux outils au panel déjà large les solutions : **Joomla** et **Drupal**, des outils assez répandus et avec des communautés actives et fertiles. Ces deux outils ont pour eux leur simplicité d'installation, de configuration et d'utilisation, ce qui a permis de créer une large base de références dans les mondes associatif, personnel et des petites entreprises.

Ces outils ont démontré leur stabilité et leur maturité. Leur simplicité, avec toutefois quelques fonctionnalités avancées, peut constituer un choix alternatif à des outils plus complets mais aussi plus complexe à mettre en œuvre.

En revanche, nous avons retiré les CMS **Cofax**, et **PHPNuke**, qui ne sont plus activement maintenus et dont les fonctionnalités sont aujourd'hui moins complètes que celles proposées par les autres solutions.

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Les autres outils ont quant à eux profité de nombreuses améliorations depuis la précédente version de ce livre blanc. Quelques exemples :

- **eZ Publish 3.9** : support du clustering, gestion multilingue poussée, Single Sign-On …
- **Typo3 4.1** : espaces de travail, contrôle d'accès front office, …
- **InfoGlue 2.5 :** Editeur d'images, webservices, nouveaux modules d'authentification, …
- **Jahia 5 :** Back-office Ajax**,** nouveaux portlets**,** Import Export XML, workflows spécifiques multi-niveaux, compatibilité JSR168, métadonnées, éditeur FCKEditor, hibernate, cache ESI, clustering, …

www.smile.fr

Les CMS

Il est maintenant acquis qu'un site web, qu'il soit intranet ou Internet, doit pouvoir être administré de manière interactive, c'est à dire qu'il doit offrir une interface permettant à ses responsables de définir de nouvelles pages, de nouvelles rubriques, de mettre à jour et réorganiser les informations.

Mais au delà des besoins propres à un site web, certaines entreprises identifient le besoin de constituer et d'administrer un *référentiel des contenus*, sur lequel s'appuieront différentes publications, tant sur le web que sur d'autres médias.

Naturellement, des outils fort nombreux sont proposés pour répondre à ce besoin. Dans l'offre de produits de CMS (Content Management Systems) on peut donc distinguer :

- d'une part les produits orientés vers la seule gestion d'un site web, simples, prêts à l'emploi, aux possibilités d'extension généralement limitées,
- d'autre part les produits plus haut de gamme, qui permettent de construire un véritable référentiel de contenus au niveau de l'entreprise, de définir les processus de gestion associés, et de décliner la restitution de ces contenus sur une variété de médias.

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Aujourd'hui le premier domaine, celui des CMS intégrés, est totalement conquis par les produits issus de l'open source, tels que Mambo, Joomla, Typo3, Drupal ou Spip. Ces produits sont d'une telle qualité et d'un tel dynamisme, pour des besoins restreints, qu'aucune offre commerciale ne peut rivaliser.

A l'inverse, le haut de gamme en matière de CMS était jusqu'à récemment le domaine de produits commerciaux tels que Documentum, Vignette ou Interwoven. Mais des solutions telles que Jahia, eZ publish et Infoglue, apportent des alternatives très sérieuses dans le domaine de l'*Enterprise Content Management*.

Depuis 2001 maintenant, Smile a bâti une expertise unique dans le domaine de la gestion de contenus, et en particulier une forte connaissance des meilleures solutions open source.

C'est la raison pour laquelle nous vous proposons ce livre blanc, qui présente un échantillon de cette expertise.

Plusieurs solutions open source assez différentes sont citées à titre d'exemple pour illustrer la manière dont sont traités les principaux concepts de la gestion de contenu.

> Soulignons que nous avons fait une sélection sévère en amont, de sorte que les produits évoqués ici constituent chacun une solution de qualité. Ce sont ensuite les spécificités de chaque projet qui orienteront vers l'une ou l'autre de ces solutions.

www.smile.fr

Les CMS open source

Un analyste du Gartner Group relevait qu'en matière de gestion de contenu, *« aujourd'hui les gens ne voient pas l'intérêt de payer les prix élevés des produits commerciaux les plus importants, et sont à la recherche d'alternatives de plus faible coût »*.

Mais, comme le souligne l'analyste John McGrath : *« les avantages des solutions open source ne se limitent pas aux économies. La gestion de contenu, par sa nature même, requiert une part importante d'adaptations spécifiques et en ayant accès au code source, les développeurs peuvent réaliser des fonctions nouvelles telles que définir un nouveau type de contenus, directement dans le code, une chose qui n'est pas possible avec des outils propriétaires. »*

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Depuis quelques années maintenant, les CMS open source sont adoptés y compris par les plus grands comptes, ainsi qu'en témoignent les références de Smile. Bien souvent, à l'occasion d'un projet de refonte, ou bien du renouvellement du contrat de support d'un produit propriétaire en place depuis quelques années, les alternatives open source sont considérées, et les avantages en particulier au plan économique, apparaissent de manière indiscutable.

Pérennité, support

Lorsque l'on choisit un produit sur lequel construire tout ou partie de son système d'information, la question de la pérennité est parfois plus importante encore que la liste des possibilités.

A cet égard, les solutions open source offrent deux avantages importants :

- Tout d'abord, leur survie ne dépend pas de leur rentabilité, ou de considérations marketing. Tant qu'une communauté s'y intéresse, le produit vivra.

- Ensuite, la libre disposition des sources, est un gage fondamental de pérennité, cela même si – il faut le souligner – il n'est nullement nécessaire que l'entreprise cliente les maîtrise.

Il faut garder à l'esprit qu'un éditeur de solutions CMS tel que REEF, qui avait été l'une des sociétés phares du marché, a coulé un jour, en laissant ses anciens clients démunis, n'ayant guère d'autre choix que de reconstruire en toute urgence leurs sites sur un autre outil. Les exemples semblables sont nombreux ces dernières années : un nom, un éditeur, une licence, même des références… ne sont pas des garanties de pérennité. La fin annoncée du pionnier NetPortal en est un nouveau témoignage.

Certes le caractère open source n'est pas en soi une garantie d'éternelle jouvence, certaines solutions peuvent lentement perdre leur communauté et leur dynamisme, mais une chose est sûre : une solution open source ne coule *jamais* de cette manière, du jour au lendemain, et de plus une entreprise qui a construit sur une solution open source conserve toujours, avec les sources, de quoi faire vivre son système au moins quelques années.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Pour ce qui est du support, les spécificités des solutions open source sont maintenant bien connues. Elles offrent un support à deux niveaux : d'une part une communauté dont l'activité s'appuie sur un site de référence, où sont disponibles les *downloads* et sources, bien sûr, mais aussi le plan d'évolutions, les bugs répertoriés, les mailing-lists de support. Ce support communautaire est non contractuel, mais néanmoins très fort. Il peut être complété du support de sociétés telles que Smile, qui lui peut être contractualisé.

A coté des open source véritablement communautaires apparaissent également des solutions qui sont le fait d'éditeurs, sociétés commerciales à but lucratif, qui ont fait le choix d'une diffusion sous licence open source. La libre diffusion du produit permet à ces éditeurs de faire connaître leur produit le plus largement possible, et le business model s'appuie sur une prestation de support et de conseil.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

SOMMAIRE

PREAMBULE 2
SMILE 2
CE LIVRE BLANC 2
QUELQUES REFERENCES DE SMILE 3
VERSION 2.1 3
VERSION 3.0 4
LES CMS 5
LES CMS OPEN SOURCE 6
PERENNITE, SUPPORT 7
SOMMAIRE 9
LES CMS OPEN SOURCE 10
MAMBO, JOOMLA, ZOPE 10
SPIP 11
TYPO3 12
OPENCMS 13
EZ PUBLISH 14
JAHIA 15
APACHE LENYA 16
INFOGLUE 17
JOOMLA 18
DRUPAL 18
CONCEPTS FONDAMENTAUX DE LA GESTION DE CONTENUS 19
GESTION DE CONTENU ET GESTION DE PUBLICATION 19
FRAMEWORK OU PRET A L'EMPLOI ? 20
SEPARATION DE LA FORME ET DU FOND 23
STRUCTURE DES ARTICLES 25
PAGES ET CONTENUS 28
ORGANISATION DES CONTENUS 30
GESTION DES CONTRIBUTIONS 33
WORKFLOW DE VALIDATION 36
WEBDAV 38
GESTION DE VERSIONS 39
CYCLE DE VIE DES ARTICLES 41
PUBLICATION 43
SELECTION DES CONTENUS 43
RESTITUTION ET GABARITS 45
PERSONNALISATION DE LA RESTITUTION 48
ECHANGES DE CONTENUS ET SYNDICATION 50
MESURE ET SUIVI D'AUDIENCE 52
INDEXATION ET MOTEUR DE RECHERCHE 53
GESTION DES URL 56
SYNTHESE 58

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

LES CMS OPEN SOURCE

Mambo, Joomla, Zope

Il existe une offre intéressante de CMS d'entrée de gamme dans l'environnement PHP. Des produits tels que **Spip**, **Mambo** ou **Joomla** répondent aux fonctions de base de la gestion de contenu, et sont à considérer pour construire certains sites.

Mambo et **Joomla** sont des outils de gestion de site web très efficaces et très simples d'accès. Comme les autres outils de ce type, ils disposent de nombreux modules prêts à l'emploi, et pratiques pour des sites de communautés : livre d'or, forum, galerie photo, blog, etc.. Leur simplicité d'accès permet de plus de créer des sites graphiquement réussis, même pour des non-informaticiens.

Sur le plan de la gestion de contenus pure, **Spip**, **Mambo** et **Joomla** restent toutefois limités, notamment sur la structuration des contenus, la classification et la transversalité de l'information, le multi-positionnement des contenus, les circuits de validation … bref, ce qui fait un outil de gestion de contenus professionnel. Il reste que ces outils, mis dans les mains d'un webmaster dynamique, seront terriblement efficace…

Ces outils PHP méritent considération, et sont des solutions appropriées pour certains sites. Nous avons du faire un tri et ne retenir que ceux qui ont un positionnement unique, c'est à dire qui constituent chacun la meilleure réponse à une catégorie de besoin.

C'est ainsi que nous avons retenu **Spip**, **Joomla**, **Drupal**, **Typo3** et **eZ publish** dans le panel traité ici, en laissant leur place également aux solutions du monde Java serveur.

Enfin, on ne peut parler de CMS sans mentionner Zope, à moins de s'attirer les foudres des nombreux *aficionados* de ce produit.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Zope est un serveur d'application très complet, qui va bien au delà de la fonction de CMS. Tout le monde s'accorde à penser que d'être construit en environnement Python est pratiquement son seul défaut. Il est vrai que les compétences sont rares pour faire du développement ou de la maintenance en Python, et cela ne risque pas de s'améliorer. C'est la raison pour laquelle nous avons choisi de ne pas retenir Zope parmi les solutions CMS privilégiées, car nous voulons être en mesure de compléter ou d'adapter les fonctions natives de l'outil lorsque cela est nécessaire. D'ailleurs l'arrêt de Zope CPS par son principal promoteur Nuxeo a confirmé l'abandon progressif de cet environnement.

Spip

Spip est un véritable CMS, avec les principales fonctionnalités que l'on peut en attendre. A l'origine, Spip était principalement utilisé par des internautes pour gérer leur site web personnel, ou pour des sites d'associations, mais Spip a grandi et est maintenant utilisé par des organismes privés ou publics, pour gérer leur site web professionnel. Spip est un projet open source français, et sa communauté peut être qualifiée d'active, avec plusieurs versions par an et des centaines – voire des milliers – de membres.

Spip est un des rares CMS à pouvoir prétendre à **plusieurs milliers de références** à travers le monde. Cette réussite s'explique par **la simplicité du produit** : à la fois la simplicité d'utilisation, mais aussi la simplicité de déploiement et d'adaptation.

En contrepartie de cette simplicité, Spip présente quelques limitations sur des fonctionnalités de gestion de contenus, ce qui le limite clairement au monde des outils de gestion de contenus orienté 'web', et non pas 'entreprise'. Notons par ailleurs que Spip dispose d'une interface d'administration entièrement multilingue, traduite dans plus de 20 langues, comprenant l'arabe, l'esperanto, le russe et même le chinois.

En parallèle à Spip, s'est développé le projet **Spip-Agora**, sponsorisé par un service du gouvernement français, avec la volonté d'ajouter des fonctionnalités au projet Spip pour l'utiliser au sein de l'administration pour des projets plus complexes.

Les fonctionnalités ajoutées, telles que l'arborescence de mots-clés ou la personnalisation du front-office, sont très intéressantes.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Le projet Agora est maintenant un projet open source distinct de Spip (un *fork*), car des choix techniques de Agora empêchent la fusion des deux projets. Agora est disponible au téléchargement sous licence GPL. La forte communauté Spip assure la pérennité du produit, mais Spip-Agora n'a jamais pu bâtir une communauté semblable, et perd donc des positions.

Smile a construit plusieurs sites avec Spip, notamment un générateur de sites Spip pour les délégations de la Croix-Rouge Française, ou encore le site du Fratel, réseau francophone de la régulation des télécommunications.

Typo3

www.smile.fr

Typo3 est le fruit de plusieurs années de travail d'un gourou danois du nom de Kasper Skårhøj. Le produit est sorti fin 2000, et depuis une communauté très active s'est développée, particulièrement dans les pays germaniques.

En termes de fonctionnalités prêtes à l'emploi, Typo3 est l'un des outils les plus riches que nous ayons trouvés à ce jour. Il offre à peu près tout ce que l'on peut souhaiter, et cela avec un niveau de finition exceptionnel. Gestion des droits et des contributions, cache, habilitations, gabarits, etc., tout y est, avec peu de limitations.

Parmi les fonctionnalités offertes par Typo3 on peut citer la manipulation d'images, qui lui permettent de redimensionner des images, de créer des vignettes et également de générer dynamiquement des titres en tant qu'images.

L'une des grandes forces de Typo3 réside dans son **extensibilité par modules**. Un module peut ajouter un ensemble de fonctionnalités à Typo3, mais aussi modifier une fonctionnalité déjà intégrée, sans modifier le code de Typo3, et donc, permet de rester compatible avec les futures versions de Typo3.

Plusieurs milliers de références opérationnelles sont indiquées sur le site de référence. En France, Smile a déjà mis en place plus de 40 sites sous Typo3, et assuré un grand nombre de formations et transferts de compétences pour aider d'autres intégrateurs.

Typo3 a notamment été choisi par Dassault Systèmes pour construire son Intranet. A cette occasion un collaborateur de Smile a eu le privilège de travailler aux cotés de Kasper sur ce projet ambitieux et passionnant.

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Smile a mis en œuvre Typo3 pour le site Internet de la CNIL, l'intranet du Château de Versailles, de l'Institut National de l'Audiovisuel, PROLEA, AmecSpie, Glamour.com, Faceo, PSA Peugeot Citroën, IPSOS, le Crédit Coopératif, Dassault Systèmes, SAS, l'AESN, CIDIL, ETS, AMUE, la Région Ile de France, Jaccede.com, ODIT France, etc.

OpenCms

OpenCms est l'outil CMS de la web agency suédoise Framfab, et en particulier de sa filiale allemande. C'est un produit d'envergure, solide, qui représente un important effort de développement.

Suite aux difficultés de Framfab, le produit semblait en panne, mais les dernières versions montrent que la dynamique de développement continue, avec un support des JSP et *taglibs* JSTL, et surtout, **la prise en compte de contenus structurés, au format Xml**. Un livre publié en anglais est disponible qui démontre encore la relative vivacité de cet outil.

OpenCms est construit sur un framework Java spécifique, dans une perspective d'extensibilité. Ainsi, même si les fonctions de base sont prêtes à l'emploi, les mises en œuvre sophistiquées induiront du développement sur le principe de l'extension des classes Java de base.

L'une des caractéristiques de l'outil est un back-office dont **le graphisme reproduit pratiquement une interface Windows** : arborescence, menus contextuels, boites de dialogues, etc. Très séduisante sur le plan esthétique, l'interface est un peu lourde à l'usage. Depuis la version 6, l'outil permet aussi d'éditer les contenus en front-office, sans passer par le back-office. Mais cet usage reste réservé à des cas très simple, et non pour l'administration complète des contenus.

OpenCms possède de belles références de sites 'corporate', dont une majorité en Allemagne.

Citons notamment le site Essilor Allemagne : www.essilor.de, le site d'un équipementier automobile : www.newtron.net, le site de Sumitomo Canada : www.sumitomocanada.com, le site de l'Unicef en Hollande : www.unicef.nl …

Smile a mis en œuvre OpenCms pour l'UDF, et pour Editions Francis Lefebvre.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

eZ publish

eZ publish se distingue des autres CMS open source par ses possibilités de configuration et d'extension : il se présente comme un outil prêt à l'emploi mais aussi comme un framework de développement pour réaliser des applicatifs métier.

Sur les fondamentaux de la gestion de contenus, eZ publish est l'un des outils les plus puissants disponibles en open source : contenus structurés, workflow, catégorisation, gestion des versions.

eZ publish permet de créer des types de contenus en ligne. En quelques clics et formulaires, il est possible de créer le type de contenu 'Compte-rendu de réunion' ou 'Actualité' ou encore 'Fiche bibliographique'. Le premier contenu sera composé d'un titre, du nom du projet associé, d'une date, d'une liste de participants, d'un corps de texte … Le second contiendra uniquement un titre et un petit texte. Le troisième sera plus complexe.

eZ publish pousse l'approche objet jusqu'au bout, en faisant de chaque contenu un objet et en lui appliquant toutes les méthodes disponibles : multipositionnement, versionning, multilinguisme, droits d'accès, workflow. Cette notion en fait un CMS hautement paramétrable.

Sur le plan technique, eZ publish est en Php, fonctionne sur Apache et MySql (PostGreSql est supporté, ainsi que Oracle et Microsoft SqlServer), met en œuvre le protocole webdav, gère trois caches différents en fichiers et sépare très distinctement contenus et présentation.

Smile a déployé eZ publish sur plus de 50 projets, incluant Bouygues Telecom, Voyages-Sncf.com, NationalGeographic.fr, Eurostar.com, l'INRA, le CIRAD, Zodiac, OSEO, Hôpital Saint-Camille, De Dietrich, Prisma Presse, Prisma Pub, Femme Actuelle, Elle, le Groupe Moniteur, Gîtes de France, CEA, Altavia, Hachette Filipacchi, GPdis, AEP, Degrémont, DxO, UCANSS, l'UDF, Merkures, CNDP, Ecureuil Gestion, Institut de l'élevage, Suez environnement, Air Pays de Loire, Amnesty International, Come and Stay, Jour du Seigneur, Souriau, Institut Français du Pétrole, National Geographic, MotorPresse, GS1, CNOP, Sport24, …

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Jahia

Jahia est un produit franco-suisse, qui tient une place à part dans le monde des portails et des CMS J2EE. Cela à deux égards.

En termes de licence tout d'abord. Il existe une version « Community », véritablement open source, mais qui n'est pas celle utilisée le plus souvent. Pour les versions commerciales, les sources sont distribuées, mais la licence n'est pas à proprement parler open source (à partir de 5 kEuros par serveur). La licence est dite 'collaborative', dans le sens où le client final peut commander des fonctionnalités nouvelles et déduire du prix de licence le coût du développement. Les développements sont alors intégrés dans les nouvelles versions du produit.

En termes de positionnement ensuite, car **Jahia est une des rares solutions qui réunit portail J2EE et gestion de contenu en un produit unique, parfaitement intégré**, et d'un prix très raisonnable.

Nous avons souhaité faire figurer Jahia dans notre comparatif CMS open source, car il constitue une excellente alternative aux solutions de portail des grands éditeurs, la possibilité de voir et de modifier les sources de Jahia assurant la pérennité et l'adéquation de l'outil aux besoins.

L'une des forces de Jahia est de réunir CMS et portail en un tout parfaitement packagé et facile à administrer. Le respect des standards techniques, notamment de la norme JSR 168, assure une réelle compatibilité et surtout la capacité à intégrer des « portlets » tiers ou spécifiques, donnant accès aux applications internes de l'entreprise.

Jahia est **prêt à l'emploi** et ne nécessite le plus souvent aucun développement dans sa mise en œuvre. Librement téléchargeable, Jahia est d'un niveau de finition et de packaging très impressionnant : il s'installe en quelques minutes, prêt à fonctionner. Ceci montre le niveau de finition de l'outil, malgré sa richesse fonctionnelle.

Jahia se distingue des autres CMS par son interface d'administration des contenus, qui est fondue dans le site lui-même. Une fois identifié, vous avez accès aux fonctions d'ajout, modification et suppression, directement sur le site, à l'intérieur des pages et des menus. Ce mode de gestion apporte une réelle simplification pour le contributeur, qui voit et comprend directement où il doit modifier l'information.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Smile a mis en place le portail Jahia pour Bouygues Immobilier, Valorissimo, Arjowiggins, Eutelsat, Beauté Prestige International, Veolia Eau, Académie de Rennes, Conseil Régional d'Ile de France, SANEF, Groupe COLAS, Fondation d'Auteuil, Académie de Créteil, IRP Auto, Pimkie-Diramode, AFNOR, OCP, Conseil Général de la Côte d'Or, Diagnostica Stago, Conseil Régional du Centre, Adoma, Chambre des Notaires de France…

Apache Lenya

Apache Lenya est un outil à part dans le monde de la gestion de contenus. Intégré à la fondation Apache, il est issu du produit Wyona CMS, d'origine suisse, qui a plusieurs sites en production à son actif (www.nzz.ch, www.unipublic.unizh.ch, www.computerworld.ch, …). La phase d'incubation préalable à l'intégration à Apache a permis de vérifier le dynamisme et le sérieux de la communauté autour de Lenya.

Lenya est un logiciel de gestion de contenus ou de documents, développé en Java, basé sur le framework Cocoon et manipulant des contenus Xml

L'une des particularités de Lenya est que **toutes les données qu'il gère sont stockées dans des** fichiers Xml. Même les informations de structure comme les groupes et utilisateurs sont gérées en fichiers Xml. Lenya ne nécessite pas de base de données pour fonctionner. Les fichiers Xml sont organisés dans une arborescence et sont affichés en Html - ou dans un autre format - au moyen de feuilles de styles Xsl.

L'interface de Lenya est entièrement web. Un simple navigateur suffit pour pouvoir créer, modifier, supprimer des documents Xml. Une fois identifié, on peut naviguer dans l'arborescence des documents, faire une recherche pour trouver le document souhaité, et effectuer les actions d'administration s'y rapportant.

Lenya permet de définir plusieurs structures de documents Xml, par l'utilisation de **schémas XSD ou RelaxNG**. On peut tout aussi bien définir un modèle strict pour une actualité (titre, date et texte) qu'un modèle large pour un document générique (comme un Word) permettant de faire des paragraphes, des titres 1, des titres 2, des encarts, ou pour définir un modèle XHTML ...

La modification des contenus Xml se fait par des éditeurs wysiwyg intégrés à l'interface web, issus de projets open source séparés : BXE, qui s'appuie sur un schéma RNG de définition des Xml, ou Kupu pour du XHTML seulement.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Smile a déjà mis en oeuvre Apache Lenya dans le cadre de trois projets de bases documentaires Xml, pour Pierre Audoin Consultants, le CIDJ et EDF. Le système mis en place permet de gérer des documents structurés, complexes, associés à des circuits de validation élaborés, et vient **en remplacement d'outils tels que Microsoft Word**. On notera que Michael Wechner, le créateur de Lenya, a collaboré avec Smile à la réalisation d'un de ces projets.

InfoGlue

InfoGlue est un outil d'origine suédoise, qui dispose de nombreux atouts :

Des atouts technologiques : InfoGlue est développé en Java, et peut utiliser indifféremment les bases de données MySql, Oracle, Microsoft Sql Server, PostGreSQL ou DB2. **InfoGlue intègre l'outil de portail open source Pluto**, ce qui lui permet d'accepter des modules d'extensions. Et **InfoGlue intègre l'outil de workflow open source de OpenSymphony**, ce qui lui permet de configurer n'importe quel workflow métier en Xml.

Des atouts fonctionnels : InfoGlue permet de configurer sans développement de nouveaux types de contenus et des catégories, deux fonctions essentielles de la gestion de contenu haut de gamme. Les autres fonctionnalités sont aussi implémentées de manières satisfaisante.

Le développement autour de cet outil est très dynamique avec plusieurs mises à jour majeures par an. La version 2.5, sortie en 2007 améliore le back office, l'interopérabilité et les performances.

Bien que le projet ait débuté en 2003, il est encore peu connu en France, mais son implémentation dans des solutions de portail, notamment pour des universités françaises, va fortement le promouvoir.

Dans le monde Java, InfoGlue est une solution qui manquait à l'offre open source. Simple d'installation, simple d'utilisation, gratuit, configurable et extensible, InfoGlue répond à de nombreux besoins, pour des sites web, mais aussi pour gérer des référentiels de contenus d'entreprise.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Joomla

Joomla est un CMS développé à partir du CMS « Mambo ». Il a été créé suite à un différent entre les développeurs principaux et la société coordinatrice des développements. Aujourd'hui, la majorité des développeurs et de la communauté se consacrent à **Joomla**, ce qui fait nettement pencher la balance en sa faveur.

Cet outil se démarque principalement par la **convivialité de son interface d'administration**. Le mot d'ordre étant de « donner un contrôle total du produit à un non-technicien ». Création de pages, catégorisation, recherche, statistiques d'accès, urls significatives ainsi que de nombreux modules sont directement intégrés et ne demandent pas de connaissances spécifiques pour leur mise en œuvre.

Cependant, Joomla ne permet pas de gérer différents types de contenus, et ne pourra pas être utilisé lorsqu'on aura un besoin fort de structuration des contenus.

Ce CMS conviendra parfaitement pour des sites personnels mais pourra également répondre aux besoins de nombreux sites professionnels.

Drupal

Drupal est un CMS aux multiples facettes. Conçu pour être un blog collectif, il trouve aujourd'hui des applications très variées : du portail communautaire au site corporate, en passant par l'intranet ou encore le site e-commerce.

La force de ce CMS est son extensibilité. Il accueille très facilement de nombreux modules complémentaires (forum, galerie photos, sondage, formulaire, newsletter, messagerie, chat, enquête, paiement en ligne, calendrier partagé, etc.).

Au niveau de la gestion de contenus, Drupal se situe à mi-chemin entre Joomla et eZ Publish. Point fort : l'interface d'administration permet de **créer facilement de nouveaux types de contenus structurés.**

Drupal intègre un système de templates PHP (très flexible), la gestion des droits utilisateurs, le multi positionnement des contenus, ainsi que des fonctionnalités de recherche et de statistiques.

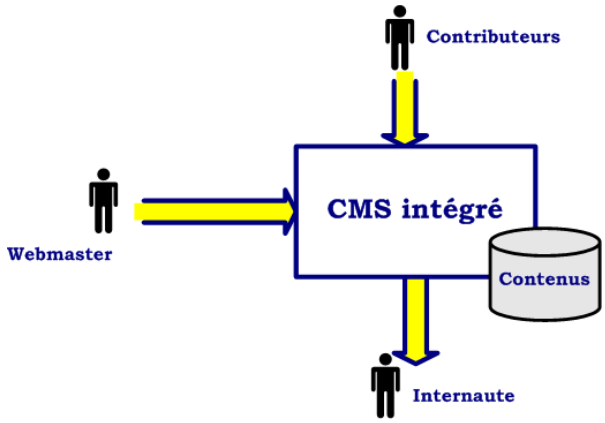
www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

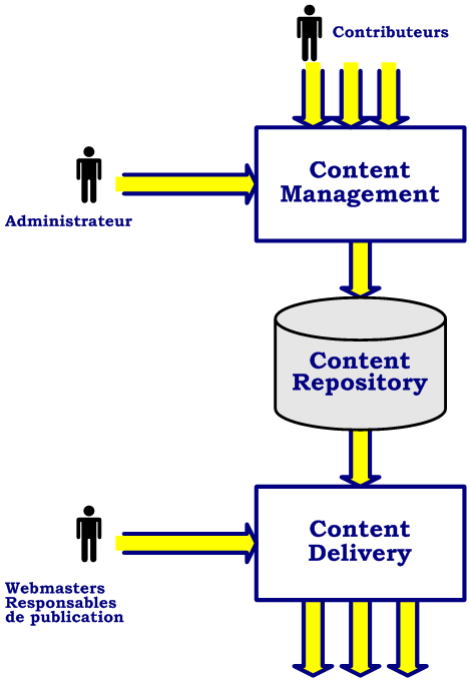
CONCEPTS FONDAMENTAUX DE LA GESTION DE CONTENUS

Gestion de contenu et gestion de publication

On réunit le plus souvent en un même outil la gestion des contenus et la gestion de la restitution de ces contenus. En anglais, ce sont les notions de *Content Management* d'une part, et *Content Delivery* d'autre part.



Les outils CMS les plus simples mêlent volontairement ces deux fonctions : leur objectif est spécifiquement de gérer des contenus destinés à constituer un site web, et leur fonctionnement est sensiblement simplifié par cette association.



Au sens strict pourtant, il convient de distinguer gestion de contenu et publication : la gestion de contenu consiste à bâtir et administrer une base des contenus, cela sans présager de l'usage qui sera fait de ces contenus. Ces contenus pourront servir à construire un site web, mais peut être aussi plusieurs sites web différents, alimentés par la même base et présentant des sélections de contenus différentes. Ou bien différents sites répondant à des chartes graphiques différentes. Mais

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

ces contenus pourront aussi être restitués sur d'autres médias : servir à la construction d'un catalogue papier, par exemple. Ou bien même la base des contenus peut être construite dans une perspective de knowledge management, sans finalité de publication.

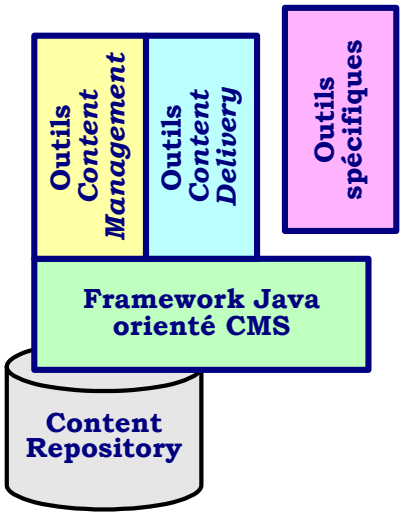
La fonction de publication, quant à elle, consiste à restituer une sélection de contenus issus du référentiel, sur un média particulier, par exemple pour élaborer un site web. Nous détaillerons les modalités de cette fonction plus loin.

Framework ou prêt à l'emploi ?

Dans le monde J2EE, certains CMS haut de gamme distinguent deux niveaux d'implémentation : d'une part un socle de composants Java spécialisés dans la gestion de contenu, et d'autre part des outils construits sur ce socle.

Cette approche offre une forte extensibilité, dans la mesure où les composants du framework sont parfaitement documentés, et peuvent être utilisés pour construire des outils spécifiques, tant pour la fonction *management* que pour la fonction *delivery*.

Par ailleurs, et c'est la force de l'environnement Java, ces frameworks sont également conçus pour que les composants de base puissent être dérivés et enrichis.



Spip est entièrement prêt à l'emploi**.** L'interface d'administration est soignée et englobe toutes les manipulations nécessaires à l'administration et à l'exploitation d'un site.

Joomla et Drupal sont également prêts à l'emploi. Ils ne constituent pas un framework, mais ils disposent d'un système de modules qui permet d'étendre les fonctionnalités si on le souhaite. Les nombreux modules disponibles permettent d'adapter ces CMS à la majorité des besoins.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

S'il est clairement prêt à l'emploi, **Typo3** a aussi comme caractéristique une architecture prévue pour accueillir des modules d'extension. Des modules déjà existants vont au delà des fonctionnalités de CMS. A noter que Smile a réalisé plusieurs de ces modules.

eZ publish est le plus souvent utilisé prêt à l'emploi, avec de plus une installation automatique disponible qui fonctionne sur de nombreuses plateformes. Il dispose aussi d'un système d'extensions, ainsi que d'un framework sous PHP.

Jahia est prêt à l'emploi, avec une qualité de packaging exceptionnelle : il est possible de le télécharger et de le faire fonctionner automatiquement en quelques minutes. Il présente néanmoins de bonnes capacité d'extensions par le mécanisme des portlets.

OpenCms est un CMS orienté framework, de sorte que les voies d'extension sont déjà identifiées et balisées. Il faut garder à l'esprit toutefois que la maîtrise d'un framework Java, même bien conçu et bien documenté, est un investissement important. Sur ce produit, il a fallu plus de 3 semaines aux ingénieurs de Smile, ayant déjà une forte expertise J2EE, pour maîtriser les concepts et la mise en œuvre de ces frameworks, et être en mesure d'y fonder de réelles extensions aux CMS.

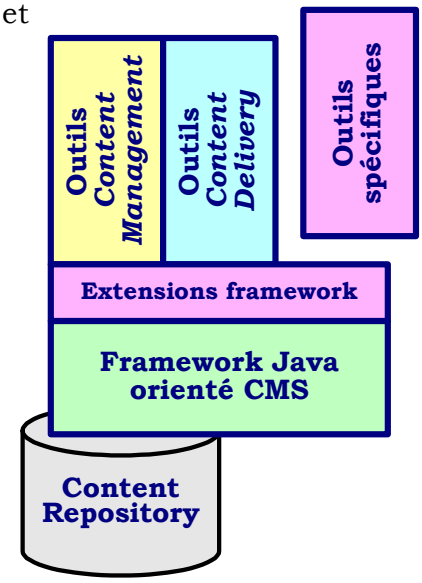
Apache Lenya est prêt à l'emploi, mais sa mise en œuvre reste très technique et relativement complexe. Lenya n'est pas fait pour accueillir des extensions, même s'il est possible d'étendre le produit en touchant le code du noyau.

InfoGlue est prêt à l'emploi, mais extensible par module. Il n'est pas orienté framework, mais peut accueillir d'autres modules ou applicatifs, par sa fonctionnalité de portail.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Notons qu'une des extensions classiques consiste à dériver des objets 'Article' standards, pour définir un objet correspondant à un type de contenu spécifique. Le développement consistera alors à redéfinir la couche persistance du nouvel objet, en référence à des tables de la base des contenus, et à définir un kit d'édition (*authoring kit*), qui gère une interface de mise à jour spécifique à ce nouveau contenu. Mais le développement d'extensions peut également s'envisager pour définir de nouveaux workflows, des cycles de vie spécifiques, ou intégrer au plus près le CMS avec des applications maison.



Notre analyse est que l'approche framework qui prévalait il y a quelques années pour le haut de gamme est aujourd'hui dépassée.

Le besoin est suffisamment stabilisé pour que les outils sachent le couvrir par simple configuration ou paramétrage. L'approche framework ou extensible reste intéressante pour des fonctionnalités annexes à la gestion de contenus, qui sont spécifiques à chaque projet. Ces fonctionnalités doivent alors être développées sans toucher au code source de l'outil, dans des modules se basant sur ce framework.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Séparation de la forme et du fond

L'un des principes des outils de CMS est d'instituer une frontière entre contenant et contenu, entre la forme et le fond, entre la présentation graphique des articles, et le texte des articles.

Il y a quantité de bonnes raisons de préserver cette séparation.

D'abord forme et fond relèvent de **métiers différents**: la mise en page est élaborée par un graphiste, tandis que le contenu est fourni par un contributeur métier.

Ensuite l'un et l'autre doivent pouvoir **évoluer indépendamment**. Il doit être possible en particulier de changer le design d'un site sans avoir à intervenir sur les contenus. Et réciproquement, d'ajouter ou de modifier un texte sans se préoccuper de sa mise en forme.

Enfin une mise en page unique est partagée par de multiples contenus: c'est ce qui fera **l'homogénéité du site**. Elle serait impossible à réaliser si chaque saisie de contenu devait définir également la mise en page.

Ajoutons aussi que la mise en forme se définit toujours en référence à **un média de publication**, de sorte que mêler mise en forme et contenus implique de réduire les possibilités de publication à un seul média.

La chose est entendue donc: d'un coté la forme, de l'autre le fond. De manière pratique, cela signifie donc que les textes insérés dans la base de contenus ne doivent pas comporter de mise en forme.

Cette contrainte est parfois trop forte pourtant, et il faut alors autoriser quelques exceptions: gras, italiques, souligné. C'est à dire le niveau de mise en forme accepté au sein d'un article de journal. Mais pas plus: fonte, taille de caractères, couleurs, alignement, agencement en colonnes, … tout cela relève de la maquette et ne doit pas être autorisé au sein du contenu.

Pour répondre au besoin de mise en forme, les CMS incluent en général un outil d'édition Html, une sorte de « mini-word », qui permet de mettre en forme les textes, mais aussi d'insérer des liens et des images. L'intégration d'un tel outil peut être plus ou moins complète, selon qu'il sache prendre en compte les styles CSS définis pour le site, que l'insertion de lien permette de sélectionner un des contenus du site ou que l'insertion d'image permette d'insérer une image de la médiathèque.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Joomla propose nativement l'éditeur HTML open source TinyMCE, éditeur wysiwyg de bon niveau qui produit du code html de qualité.

Drupal n'intègre pas nativement de mini-word, mais des modules complémentaires permettent de rajouter facilement cette fonctionnalité, en utilisant les éditeurs open source les plus répandus. L'intégration de l'éditeur externe est alors simple, par exemple sans gestion des liens vers les contenus.

OpenCms dispose nativement d'une solution d'édition Html, bien intégrée à l'outil.

Jusqu'ici **Typo3** avait opté pour une solution VBscript qui ne fonctionnait que sous InternetExplorer. Dans sa version 4, Typo3 intègre maintenant l'éditeur wysiwyg HtmlArea, ouvrant ainsi la compatibilité aux navigateurs Firefox. Cet éditeur est parfaitement intégré à Typo3, avec gestion des images et personnalisation possible de l'éditeur selon les utilisateurs.

Concernant **Spip**, il est à remarquer qu'il n'y a pas d'éditeur HTML wysiwyg et qu'il est remplacé par des balises spécifiques que l'utilisateur doit connaître. Ceci limite beaucoup les possibilités d'édition des pages, comme pour inclure des images dans le texte, ou un tableau ou une structure plus complexe. Il reste toutefois possible d'intégrer un éditeur wysiwyg tel que FckEditor à Spip. Notons cependant que l'absence d'éditeur impose des mises en formes simplifiées, facilement séparables du texte du contenu, notamment lors d'export au format Xml.

Jahia intègre plusieurs éditeurs web, ce qui assure ensuite un fonctionnement sur toutes les plateformes client.

eZ publish intègre depuis peu son propre éditeur "Online editor", récemment reversé en open source sous licence GPL. Cet éditeur wysiwyg écrit en fait du XML au lieu de html, ce qui impose certaines limitations dans la mise en forme, mais assure une parfaite séparation de la forme et du fond. De plus, parfaitement intégré au CMS, cet éditeur permet d'insérer des fichiers issus de la médiathèque ezpublish, d'uploader un fichier directement en fichier joint du contenu.

Apache Lenya propose nativement deux éditeurs wysiwyg. Kupu qui permet de faire du xHtml, et BXE qui permet de modifier des contenus Xml. BXE va plus loin que la simple mise en forme. BXE permet de créer un contenu visuellement, tout en travaillant aussi sa mise en forme, selon les limites imposées par le schéma RNG. Pour le besoin de projets, Smile a aussi intégré l'éditeur Xml Xopus, outil propriétaire et payant, qui s'appuie sur des schémas XSD.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

InfoGlue intègre nativement un mini-word, basé sur FCKEditor, très bien intégré au CMS, et qui permet de créer un lien vers un contenu en le sélectionnant, de même que pour les images.

La bonne séparation de la forme et du fond est en fait davantage une question de politique qu'une question d'outils.

Sur un CMS intégré tel que **Spip, Joomla, Drupal, OpenCMS ou Typo3**, il n'est pas déraisonnable d'autoriser les contributeurs à définir des mises en forme au delà du gras et de l'italique. La contrainte du multi-canal est moindre, et l'on peut compter sur le processus de validation pour contrôler ces contributions et assurer l'homogénéité des pages.

A l'inverse, dans une optique de CMS d'entreprise, tels que **eZ publish, InfoGlue, Lenya** ou **Jahia**, cette contrainte devrait être respectée avec rigueur. Il faut souligner que les possibilités de structuration de types d'articles spécifiques limitent sensiblement le besoin d'insertion de mise en forme dans les articles, puisque pour chaque champ d'article une mise en forme pourra être définie au niveau des *gabarits*, comme on le verra plus loin.

www.smile.fr

Structure des articles

Un article, c'est un élément de contenu. Selon les cas, il peut représenter entre quelques lignes et quelques pages de texte, traitant d'un sujet donné.

Un article n'est pourtant pas la plus petite unité de structuration du contenu : un article est lui-même généralement décomposé en champs élémentaires, qui peuvent être par exemple *titre, sous-titre, image, corps de texte*.

On peut distinguer deux manières de structurer ainsi les articles : une structuration fonctionnelle et une structuration sémantique.

La structuration fonctionnelle va décomposer l'article sur la base du rôle, de la *fonction*, assurée par chaque item, typiquement *titre, sous-titre, résumé*. Cette fonction est indépendante du sujet traité, du thème de l'article : que l'on parle de recette de cuisine ou de match de foot, on peut identifier un champ titre et un champ résumé.

La structuration sémantique décompose l'article sur la base de la signification de chaque item. Pour une recette de cuisine, on pourra distinguer des champs *ingrédients, temps de cuisson, température du*

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

four, niveau de difficulté, etc. Pour des match de foot : *équipes, stade, date et heure*, etc. Comme on le voit, la structuration sémantique dépend étroitement du thème de l'article, de sorte qu'au sein d'une même gestion de contenu, on sera amenés à définir plusieurs structures ou *types d'articles*. Chaque contenu sera d'un type ou d'un autre, et il ne sera pas toujours possible de restituer un article d'un type A là ou un type B était attendu.

Le fait de bien structurer chaque article de manière spécifique présente des avantages déterminants :

- Cela permet d'une part de bien **guider la saisie** des articles au moyen d'un formulaire adapté, et donc de s'assurer qu'ils entrent bien tous dans le moule correspondant à leur type. C'est donc un facteur de **qualité** des contenus.

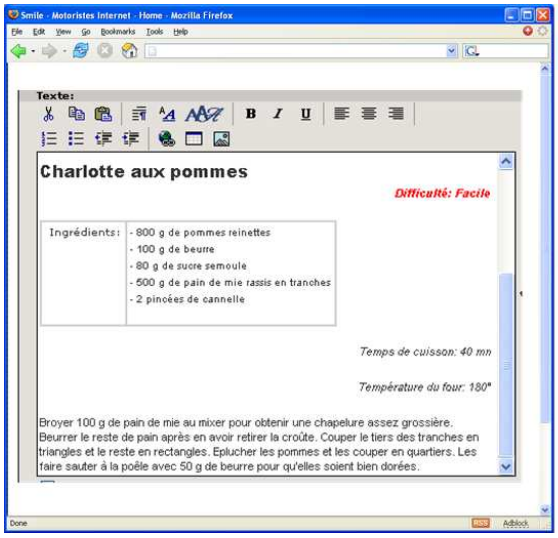
- Cela permet également de distinguer, lors de la publication, chacun des composants de l'article, et donc de **véritablement séparer la forme du fond**, telle que nous l'avons évoqué plus haut. Il sera possible de définir dans la maquette de restitution de l'article, que le champ *ingrédients* est à présenter en italique, tandis que le *temps de cuisson* est en gras. Et l'on pourra effectivement modifier cette restitution sans toucher aux recettes de cuisine. En l'absence de structure d'article adaptée, il aurait fallu soit renoncer à ces effets sur les ingrédients et le temps de cuisson, soit injecter de la mise en forme dans le corps de l'article.

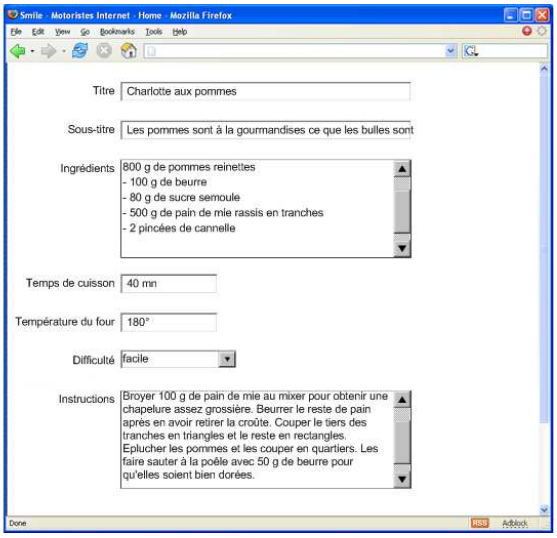
- Enfin, c'est ce qui **donne du sens aux contenus**. Les contenus ne sont plus des champs de texte, ils sont des attributs qualifiant un objet : *pays, ville, prix, date, etc…* et ce sont donc toutes les possibilités d'**utilisation de ces contenus** qui sont décuplées.

On voit donc bien l'intérêt qu'il y a dans cette manière de gérer les structures d'article. Mais il faut bien mesurer que ces avantages sont obtenus au prix d'**une certaine complexité**. Le CMS devient alors un véritable SGBD : on définit de nouvelles entités, les attributs de ces entités, les contrôles à effectuer sur chacun des champs, les relations entre ces entités, etc… Néanmoins, lorsque la déclaration d'un type de contenu structuré peut se faire sans développement informatique, alors **c'est une des fonctionnalités les plus importantes d'un CMS**.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation





Sur l'image précédente, à gauche la « Charlotte aux pommes » version non-structurée, et à droite la même en version structurée. On conçoit aisément que la maîtrise de la restitution comme de l'utilisation des contenus est beaucoup plus grande en présence de contenus structurés.

Joomla et **Spip** ne permettent pas de définir des types d'articles spécifiques. Il s'ensuit quelques limitations, mais aussi une grande simplicité dans la gestion et l'organisation d'un site. Tous les articles se saisissent sur un même formulaire, tous les articles peuvent être restitués sur un même gabarit.

Drupal supporte nativement les contenus de types différents, mais ne propose pas de gérer directement leur édition. Un module optionnel rajoute cette possibilité et permet de créer simplement de nouveaux types de contenus à l'aide de formulaires intégrés au back office.

Le module TemplaVoila de **Typo3** permet de définir de nouvelles structures de contenus, avec des formulaires de saisie adaptés. Si la création reste un peu technique, le résultat est bien là, on peut créer de nouveaux types de contenus, et les utiliser simplement partout dans le site. Les contenus structurés sont stockés en Xml dans un nouveau champ de la table des contenus.

Depuis la version 6, **OpenCms** permet de définir des types d'articles spécifiques, basés sur des schémas XSD. Les structures possibles restent simples et n'utilisent pas tout le potentiel du Xml, mais il est possible de créer simplement de nouvelles structures de contenus. Les contenus structurés sont ensuite stockés en Xml, dans un champ de la base de données.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

eZ publish se distingue dans ce panel. C'est l'outil qui permet le plus aisément de définir en ligne tous les types de contenus que l'on souhaite, tout simplement via l'interface web. Par défaut, les nouveaux types de contenus ont un gabarit standard qui affiche tous les champs du contenu. Pour chaque type de contenu, il faut ensuite définir un gabarit de prévisualisation dans l'interface d'administration et un autre pour l'interface de restitution.

Jahia permet de définir des nouveaux types de contenus par le biais de conteneurs JSP dans lesquels il suffit de définir la structure du contenu. Il n'y a pas d'interface graphique pour créer des nouveaux types de contenus, mais la création reste simple et se fait au sein d'un fichier JSP.

InfoGlue permet de créer de nouvelles structures de contenus via son interface, sans connaissance technique nécessaire. Et InfoGlue va même plus loin, en étant le seul CMS permettant de définir des règles de validité pour chaque champ, en utilisant les expressions régulières. Une expression régulière va nous permettre d'imposer que la première lettre soit en majuscule, ou qu'il n'y ait pas d'espace, ou encore que le champ soit un entier suivi d'une lettre …

Lenya, parce qu'il est basé entièrement sur du Xml, permet de définir des structures de contenus complexes. Si la définition de ces structures se fait par un schéma XSD ou RNG, donc par des techniciens, l'utilisation de ce schéma permet d'aller beaucoup plus loin dans les structures possibles. Le format Xml permet d'inclure des listes d'éléments structurés dans un contenu, ce que ne permet pas la structuration simple des autres CMS.

www.smile.fr

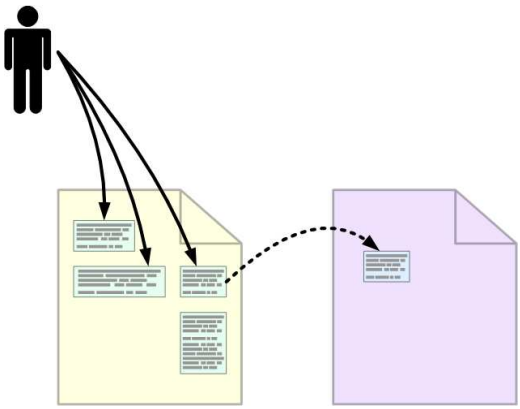
Pages et contenus

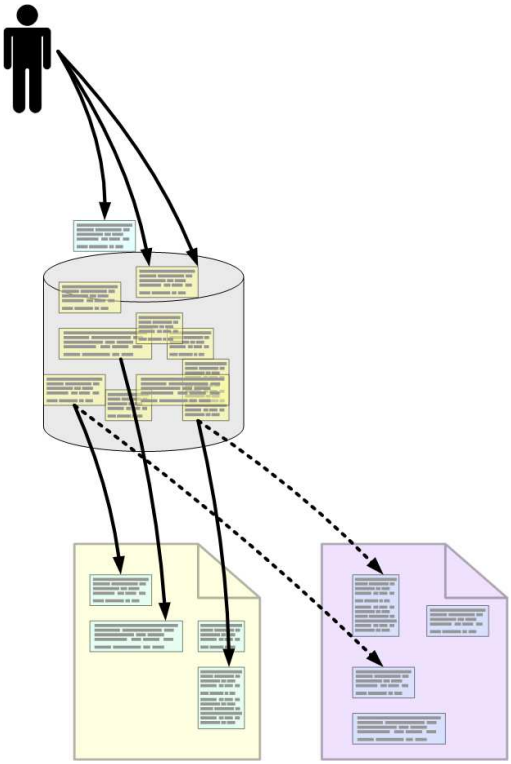
Le CMS manipule des *contenus*, souvent appelés *articles*. Ces contenus sont publiés dans des *pages*. Une page peut contenir différents articles, et un même article peut apparaître dans différentes pages.

Pour certains CMS, un contenu est toujours créé *dans une page spécifique*, voire même *à une position spécifique dans la page*. Pour d'autres, la création d'un contenu, sa validation, son versionning, sont indépendants de la notion de page. Ce n'est qu'une fois créé et validé que le contenu peut être *placé* dans une ou plusieurs pages. Cette approche découle du principe de séparation de la *gestion* en amont et de la *publication* en aval.

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Même lorsqu'un contenu est placé *en premier lieu* dans une page, il est important qu'il puisse être restitué dans d'autres pages sans pour autant être dupliqué. Ainsi une mise à jour du contenu impactera automatiquement les différentes pages dans lesquelles il est publié.





Cette restitution multiple d'un contenu, parfois appelée *multi-positionnement* des articles, peut être opérée par un positionnement explicite : le gestionnaire place l'article dans telle et telle pages, ou bien elle peut s'envisager au travers de *règles* régissant la publication.

Typiquement la présentation des cinq dernières actualités en page d'accueil, ou bien en marge de certaines pages est un exemple de publication par règles. Il n'est pas nécessaire qu'une nouvelle actualité soit *placée* dans telles et telles pages, c'est parce qu'elle est de type *actualité* et parce que elle est récente, qu'elle y figurera automatiquement.

Une différence forte entre les produits s'observe au niveau du processus de validation. Si ce processus se fait au niveau de chaque contenu, alors le CMS sera orienté 'contenus', voire 'référentiel de contenus'. Si ce processus se fait au niveau de chaque page, elle-même incluant un ou plusieurs contenus, alors le CMS sera orienté 'pages', permettant en général plus de maîtrise graphique de la page par les contributeurs.

Spip, Drupal et **Joomla** sont orientés 'contenus', chaque contenu (article, brève, rubrique, …) formant une page, avec une url. Les mots-clés permettent ensuite de ventiler les contenus sur plusieurs pages.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Typo3 est orienté 'pages'. Une page contient une ou plusieurs colonnes, et chaque colonne contient un ou plusieurs contenus ou modules, ce qui donne une grande souplesse de mise en page. La validation et le suivi des versions se font au niveau de la page dans son ensemble.

Jahia est lui aussi orienté 'pages', reflétant là son aspect portail. La mise en forme des pages est souple, dépendant des gabarits réalisés. La catégorisation bien implémentée – ou encore le 'piqueur de contenus' - permet ensuite de ventiler les contenus sur plusieurs pages. La validation et le suivi des versions sont réalisés par pages, mais depuis la version 4.5 la validation est également gérée au niveau de chaque contenu.

eZ publish est davantage orienté 'contenus'. Les contenus suivent le processus de validation, sont suivis en versions, sont placés dans plusieurs dossiers, qui sont eux-mêmes des contenus. Un contenu a sa propre url. La gestion des pages se fera au travers des templates, ou de l'onglet 'design', qui permet d'insérer des modules dans des endroits spécifiques de la page.

InfoGlue a une orientation 'contenus'. Les contenus sont validés, suivis en versions et multi-positionnés. La gestion des pages correspond à la gestion des gabarits, paramétrables en wysiwyg.

Apache Lenya a une orientation purement 'contenus'. C'est d'ailleurs son rôle, dans un mode de 'gestion documentaire' ou encore de 'production documentaire', le contenu est placé au centre, et l'affichage n'est que secondaire.

www.smile.fr

Organisation des contenus

Le mode d'organisation le plus classique est bien sûr celui d'une structure hiérarchique arborescente, semblable à la structure de répertoires d'une arborescence fichier. Chaque niveau d'arborescence, chaque division en répertoires, correspond à une logique, un classement par rapport au sens de chaque item, par exemple en référence à des thèmes.

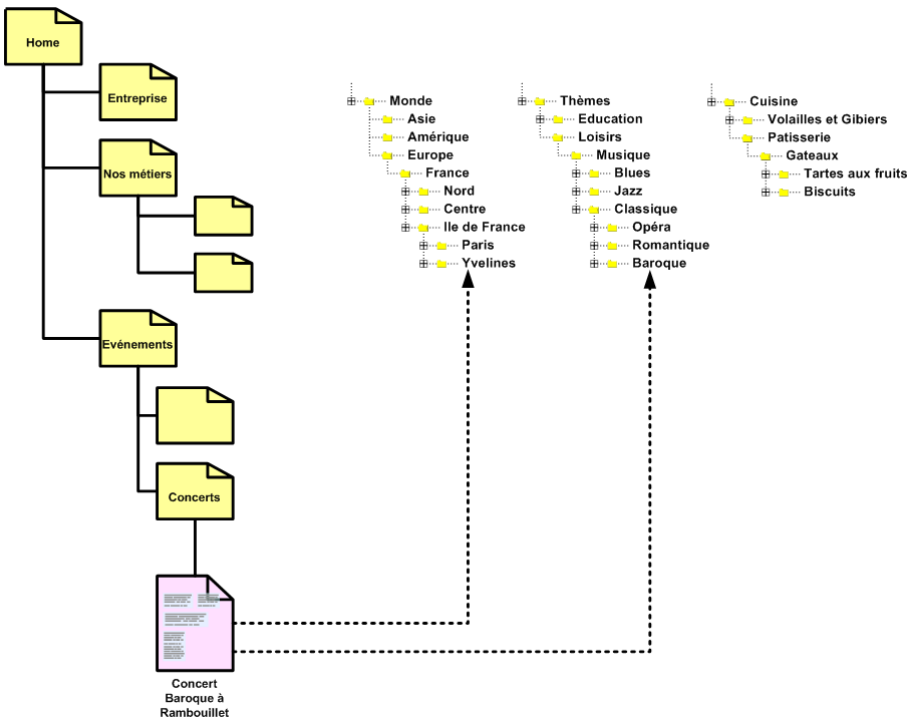
Ainsi par exemple on pourra organiser le contenu d'un site de media en *sport, culture, international, …*, puis diviser la rubrique *sport* en *basket, football, ….*

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Dans cet exemple, l'organisation des contenus est en correspondance avec l'organisation de la publication. Ce n'est pas une obligation, mais on comprend bien que cette correspondance entre organisation interne et organisation externe simplifie considérablement le travail du gestionnaire.

Pourtant lorsque la gestion de contenus s'envisage au niveau de l'entreprise comme socle de plusieurs publications, il est fort possible que l'organisation des publications ne soit pas l'image de l'organisation du référentiel des contenus. Dans ce cas, il convient de définir de quelle manière les articles viennent se placer dans la structure de chaque publication, soit **de manière explicite**, en positionnant chaque article, soit **au moyen de règles de gestion**.

Même au sein d'un unique site, **l'organisation hiérarchique pure n'est pas toujours satisfaisante**, et peut être trop structurante. Il y a des alternatives. On peut par exemple associer un article à différents thèmes, ce qui crée une organisation plus relationnelle, ou ensembliste, que hiérarchique. C'est ce que l'on appelle parfois la **catégorisation** des contenus, qui est une fonctionnalité importante de la gestion de contenus.



Le classement non strictement hiérarchique est souvent une nécessité au niveau de la publication. L'article traitant d'un match de foot de l'équipe locale pourra être trouvé à la fois dans la rubrique 'sport' et dans la rubrique 'infos locales'. **La possibilité de retrouver**

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

un article en différents points de l'arborescence permet de satisfaire des visiteurs qui ont chacun une logique particulière. En effet, et c'est un point fondamental, il ne suffit pas de présenter son contenu de manière logique et organisée, il faut aussi **que cette logique soit celle du visiteur**. Or tous les visiteurs ne raisonnent pas à l'identique, et il faut donc aussi permettre de retrouver un contenu selon différentes logiques.

Pour **Spip**, un article appartient à une et une seule rubrique. Mais on peut lui associer plusieurs mots-clés, et l'organisation par mots-clé permet de définir d'autres arborescences, limitées à 1 sous-niveau. La structuration du site est donc possible et assez souple**.** Toutefois, la question du tri (des sections, des articles) dans l'interface d'administration, n'est pas du tout abordée en dehors du tri automatique par date.

Dans **Joomla**, le site se décompose en sections, qu'il est possible d'atteindre via le menu principal. Les sections possèdent un certain nombre de catégories, auxquelles un article peut être associé. Un article possédant une et une seule catégorie, il n'est pas possible de multi positionner un contenu.

Drupal utilise un système de catégories pour organiser les contenus dans le site. Ceux-ci peuvent être multi-positionnés en les faisant appartenir à plusieurs catégories à la fois.

Typo3 mêle quelque peu la structure de restitution et l'organisation des contenus. Ainsi, un article est défini *au sein d'une page*. Une page peut bien sûr contenir plusieurs articles, et pourra être attachée en plusieurs points du site éventuellement, mais un article, lui, n'est pas mappé dans plusieurs sections. Il est possible de restituer un même article sur une autre page, mais cette restitution est toujours à définir explicitement, article par article. C'est pourquoi Smile a réalisé une extension « Catégorisation », qui enrichit le produit sur ce plan.

Jahia met en œuvre un système de classification transverse, qui permet de définir autant de structures arborescentes que l'on souhaite. Un contenu est rattaché à une ou plusieurs classifications, qui n'ont pas de limite de profondeur.

eZ publish permet de placer un contenu à plusieurs endroits du site, en gardant une référence unique. Ce multi-positionnement se fait pour tous les contenus, y compris les rubriques, ce qui permet de placer une rubrique entière et toute sa sous-arborescence à plusieurs endroits d'un site. Enfin, il suffit que certaines des arborescences de positionnement ne soient pas publiées pour mettre en œuvre la fonctionnalité de catégorisation des contenus.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

InfoGlue propose deux méthodes de rangement : soit on utilise le multi-positionnement, qui permet de placer un contenu à plusieurs endroits, et même de le réutiliser sur différents sites, soit on utilise l'arbre de catégorisation qui, appliqué aux contenus permet de définir de nouvelles navigations du site.

Apache Lenya ne propose pas nativement de méthode de classification. Pour Lenya, toute l'information est dans le document Xml, aussi sa catégorisation doit aussi y être. Dans la même logique, un document Xml ne peut pas être à deux emplacements simultanément. Pour organiser, il faut définir la classification possible dans le schéma XSD ou RNG, ce qui permettra ensuite aux contributeurs de classifier le document lors de la saisie, et de réutiliser ces catégories dans les gabarits.

www.smile.fr

Gestion des contributions

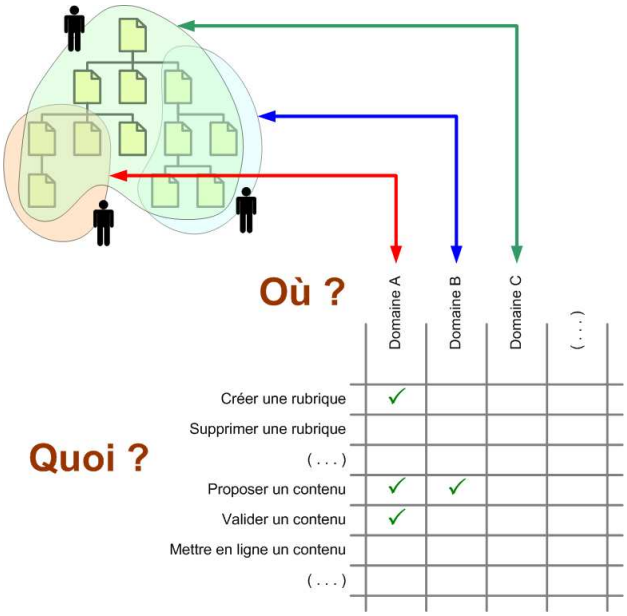
Les sites de petite et moyenne envergure ont souvent un contributeur unique, le webmaster. Qu'il rédige lui-même les contenus, ou plus couramment, qu'il les collecte au sein de l'entreprise, il est du moins le seul à introduire des contenus dans le site. Il s'ensuit bien sûr qu'il est habilité à saisir ou modifier les contenus en tout point du site.

Sur des sites de grande envergure, il n'est plus possible qu'une seule personne ait la charge de saisir tous les contenus. De plus lorsque les contenus sont en fait rédigés en dehors du CMS, ils doivent être ressaisis, ou bien copiés / collés, dans le CMS par le webmaster, ce qui représente une étape inutile.

Pour toutes ces raisons, **il peut être utile de faire directement interagir les contributeurs premiers sur le CMS, en leur offrant une interface pour la saisie, puis la modification, de leurs articles**. De plus, dans le contexte d'un Intranet, la contribution des collaborateurs est un facteur de participation, et donc d'adhésion : l'Intranet n'est plus un outil de communication 'descendante', mais devient un outil d'échange.

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Dans ce contexte, il sera le plus souvent nécessaire d'accorder aux contributeurs des **droits limités à certaines parties du site**. Ces droits sont bien sûr définis en référence à l'organisation hiérarchique des contenus.



On définit ainsi par exemple une rubrique 'Ressources Humaines', qui peut être le point de départ d'une importante arborescence de contenus, et l'on accorde à certains contributeurs le droit de saisir des articles relevant des Ressources Humaines. Les mêmes contributeurs ne pourront pas saisir des articles relevant de la rubrique 'Produits et Services'.

Les droits de chacun sont donc définis *en référence à l'organisation des contenus*. Mais doivent également être définis *en référence aux actions possibles sur ces contenus*. Ainsi tel intervenant pourra *proposer un article*, mais ne pourra pas le valider ou le mettre en ligne. Tel autre pourra valider un article dans les rubriques dont il est en charge, mais ne sera pas autorisé à créer de nouvelles rubriques, etc.

Les CMS doivent donc offrir une gestion d'habilitation à deux axes : le premier est l'axe des contenus, et de leur organisation, le second l'axe des fonctions liées à la gestion de ces contenus : consulter, mettre à jour, valider, publier, etc…

Spip permet à plusieurs contributeurs d'agir sur le site, avec des droits différents. Pour Spip, il s'agit d'<auteurs> , qui sont soit rédacteurs soit administrateurs, et qui accèdent à toutes les rubriques ou à une seule. Spip permet aussi à n'importe quel internaute de participer à un forum autour d'un article. Cette fonctionnalité, souvent mise en oeuvre dans les sites web des communautés, permet de faire vivre un site au travers des internautes, et apporte un réel 'plus' à un site éditorial. La modération du forum se fait à priori ou à posteriori.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Joomla propose 3 grand niveaux de contribution : « author », « editor » et « publisher » qui permettent respectivement de modifier, créer et publier du contenu. Chaque article peut être associé à un auteur donné par un utilisateur disposant de droits supérieur. Dans le cadre d'un simple site web, ce système est amplement suffisant.

Drupal dispose d'une gestion très fine des droits. Il est possible de créer un nombre quelconque de rôles et de leur affecter des droits pour chaque fonctionnalité de chaque module. Ce modèle permet à Drupal de s'adapter à n'importe quel besoin.

Typo3 possède une gestion des droits des contributeurs bien plus complète que celle de Spip. Comme OpenCms, elle est inspirée du modèle Unix, avec les notions de *user, group, all.* **Spip** et **Typo3** offrent l'un et l'autre une combinaison de la navigation en simple visiteur et de la contribution : à partir du moment où le visiteur est identifié comme contributeur, il peut naviguer sur le site et chaque page est accompagnée de liens permettant sa mise à jour. C'est une manière très conviviale de gérer l'interface de mise à jour.

OpenCms propose une gestion des droits très inspirée de Unix : chaque section de l'arborescence est vue comme un répertoire, et les droits de *lire, écrire, voir* sont attribués par rapport aux périmètres croissants : *user, group, other*, à savoir celui qui a écrit l'article, ceux de son groupe, et les autres.

eZ publish permet de définir des rôles, des groupes et des utilisateurs. Les rôles comprennent l'ensemble des droits autorisés, pour chaque module, en lecture ou écriture. Ce même système permet de restreindre les droits en consultation sur le site, ou en écriture sur un forum. Il permet également la contribution « in-site », c'est à dire intégrée au site, ce qui est généralement apprécié des contributeurs occasionnels.

Jahia permet de définir les profils et les droits nécessaires à une gestion décentralisée, avec les notions de groupes, utilisateurs, manipulables dans tous les sens . La gestion des droits est aussi appliquée aux modules du portail, ce qui permet d'avoir une identification unique pour l'ensemble des applications mises en œuvre.

InfoGlue permet de définir des groupes et des rôles, associés aux utilisateurs, qui peuvent donc accéder à certains modules, certains contenus et certaines actions uniquement.

Lenya gère aussi de manière fine les droits de contribution, avec un atout au niveau du circuit de validation et des rôles des utilisateurs.

www.smile.fr

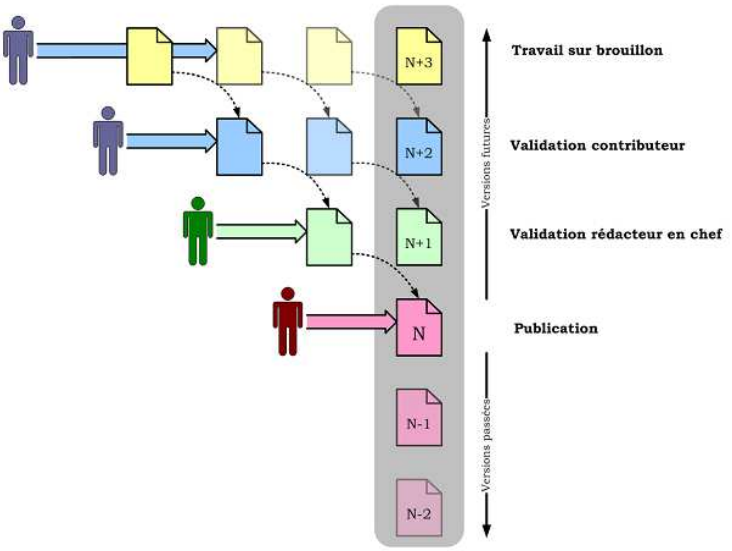
© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Workflow de validation

Le besoin apparaît souvent de pouvoir séparer les étapes de contribution et de validation. C'est le cas en particulier dans des contextes où la contribution est décentralisée, déléguée à un nombre important d'intervenants. Si l'on souhaite malgré tout garder un contrôle sur les contenus, et c'est souvent fondamental, il faut qu'un article soit d'abord rédigé, puis dans un second temps validé et éventuellement mis en ligne. Cela est d'autant plus important que les contributeurs ne sont pas toujours des spécialistes, ni de la communication, ni du web.

Dans certains contextes, ce sont même plusieurs échelons de validation qui sont nécessaires: un contributeur écrit, son chef de service valide à son niveau, et le responsable de la communication valide une seconde fois.

La validation doit donc être mise en œuvre par le CMS avec toutes les caractéristiques d'un véritable workflow: les intervenants doivent être avertis par mail des tâches qui leur incombent, et doivent d'un simple clic pouvoir consulter la liste des articles qui attendent leur validation.



Mais ce n'est pas tout, une fois un contenu créé, validé puis publié suivant le circuit de validation, il doit pouvoir être modifié, puis validé puis publié, pendant que ce contenu en ligne reste inchangé. A un circuit de validation, il faut donc pouvoir associer une gestion avancée des versions, permettant d'avoir plusieurs états simultanés pour un même contenu.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Avec **Spip**, un article peut être à l'état *brouillon*, *validé*, *publié*, *refusé* et *supprimé*. Selon son statut de rédacteur ou d'administrateur, on pourra ou ne pourra pas mettre en ligne un article. Le workflow mis en place est donc assez limité, mais suffisant dans bien des cas. Notons que pour modifier un article déjà en ligne, il faudra le dupliquer pour passer ensuite par le circuit de validation, un contenu Spip ne pouvant avoir qu'un état à la fois.

Pour **Joomla** et **Drupal**, la gestion de la validation est très simple. Un contenu est dans un état « non publié » ou « publié ». Seul un utilisateur disposant de suffisamment de droits pourra passer un document dans l'état publié. A noter que **Drupal** dispose d'un module qui implémente un workflow classique (brouillon, validé, publié).

Typo3 intègre depuis sa version 4 la notion de *workspaces* qui permet de travailler sur plusieurs environnement (une 'préprod', une version 2008, une recette, ...) puis de basculer en production une page ou une arborescence de pages.

Pour chaque workspace, on définit trois groupes : contributeurs, relecteurs et valideurs, correspondant donc à un workflow de publication à 3 niveaux, spécifique au workspace. Au final, le valideur décide de la publication dans le workspace « live ».

OpenCms, de son coté, offre un dispositif de workflow qui est à la fois extrêmement générique, et pratiquement décorrélé du processus de validation proprement dit : il permet de construire toute sorte d'enchaînement de tâches, mais il n'y a pas de lien précis entre ces tâches et la gestion de contenu. L'avantage est que l'on peut construire ainsi un workflow dans lequel on demande à tel intervenant de compléter tel article, ou de ranger son bureau. L'inconvénient est que la valeur ajoutée de l'outil est faible par rapport à un simple échange de mail.

eZ publish permet de définir des workflows associés à des sections, appliqués ou non selon les utilisateurs et groupes. Cette fonctionnalité permet de mettre en oeuvre des workflows à un ou plusieurs niveaux de validation. Un système de notification envoie automatiquement des mails d'alertes et de demandes de validation aux intéressés. De plus, eZ permet d'avoir simultanément autant de versions que l'on souhaite d'un même contenu : 3 brouillons, 1 en cours de validation, 1 publié, 5 archivé ... ce qui évite tout conflit dans les mises à jours par les différents intervenants.

Jahia, depuis sa version 5.0, permet de mettre en œuvre des workflows multi-étages et de les affecter aux contenus selon une logique d'héritage. Un nouveau workflow peut être défini de manière interactive, au sein du back-office.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

InfoGlue intègre l'outil de workflow open source OpenSymphony, qui permet de définir en Xml des workflows complexes. Ces workflows paramétrables peuvent être appliqués aux contenus, mais aussi être couplés à d'autres applications de l'entreprise, par le biais de web services. La gestion des versions est ici bien traitée, avec d'une part la possibilité de modifier un contenu et de suivre le workflow en laissant le contenu en ligne inchangé, et d'autre part la possibilité de publier des contenus par paquets, ce qui permet d'obtenir des versions de sites cohérentes, publiables et dépubliables.

Dans **Lenya**, il est possible de définir un workflow en Xml, avec autant d'étapes que souhaité, et de l'enrichir avec des étapes de transition conditionnelles. La syntaxe utilisée est relativement simple. Lenya gère deux espaces : *authoring* et *live*, ce qui permet de travailler sur la version dans *authoring*, sans modifier la version dans *live*, jusqu'à la validation, qui met à jour le contenu dans *live*.

www.smile.fr

Webdav

Le protocole http s'est enrichi d'une surcouche permettant de faire d'un serveur web un serveur de fichiers. Le WEBDAV (Web-based Distributed Authoring and Versioning) permet à des utilisateurs d'accéder et de modifier des fichiers sur des serveurs web distants.

Par rapport au FTP bien connu, le webdav permet en plus de verrouiller les fichiers lorsqu'ils sont modifiés par un utilisateur, mais surtout, il évite de devoir faire une copie locale du fichier avant de le modifier. **Les utilisateurs peuvent alors travailler collaborativement sur les fichiers du serveur distant, comme s'il s'agissait d'un simple répertoire partagé sur le réseau.**

Les postes des utilisateurs doivent être équipés d'un client webdav leur permettant d'y accéder. L'explorateur de fichiers de Windows en est équipé, ainsi que Microsoft Word, et il existe plusieurs clients disponibles (et gratuits) pour Linux ou Mac.

L'intégration de webdav aux outils de gestion de contenus ouvre de nouvelles fonctionnalités, notamment pour l'accès aux fichiers publiés, ainsi que dans le processus de publication.

Notons que l'accès webdav à un CMS doit mettre en œuvre la même sécurité que l'accès interface web, avec les mêmes utilisateurs et droits.

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Spip, Joomla et **Drupal** ne dispose dans d'accès webdav.

Typo3 prévoit le support de webdav pour sa future version 5.0, pour l'accès au répertoire fileadmin. L'intérêt de ce webdav reste cependant restreint car seuls les fichiers de la médiathèque seront servis via le webdav, et sans gestion de version.

OpenCms n'intègre pas de fonctionnalité webdav, ni n'en prévoie.

eZ publish met en œuvre le webdav de manière très spécifique. En effet, eZ publish donne accès à l'arborescence des contenus en webdav, avec une arborescence de répertoires et de fichiers reproduisant l'arborescence des contenus. Si un contenu est de type fichier dans eZ, alors il sera accessible directement en fichier via un client webdav. Cette représentation webdav des contenus permet de plus une reprise simple de contenus nombreux, par simple glisser-déplacer d'une arborescence complète de fichiers.

Jahia intègre le protocole webdav pour l'accès aux ressources fichiers associées aux contenus, et permet l'enregistrement simple de fichiers dans le répertoire des ressources, pour les lier ensuite aux contenus.

InfoGlue ne propose pas de fonctionnalité liée au webdav.

Lenya donne accès à ses contenus en webdav, pour être modifiés avec un éditeur Xml ou XHTML local. Les opérations de publication restent à réaliser sur l'interface web.

www.smile.fr

Gestion de versions

La gestion de version et la capacité à conserver les versions antérieures d'un même article, font partie des fonctions de base d'un CMS. Elle s'applique à la fois aux versions anciennes, et aux versions futures. Un même article peut ainsi avoir une version en ligne, une version N+1, validée mais non publiée, et une version N+2 à l'état de brouillon.

Comme nous l'avons vu précédemment, la possibilité d'avoir plusieurs versions simultanées d'un même contenu est nécessaire au fonctionnement des workflows de validation. C'est ce qui nous permettra de suivre des circuits de validation pour chaque modification, et donc de toujours s'assurer de la qualité des publications par un responsable.

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

La gestion des versions peut se faire au niveau de chaque contenu, mais aussi au niveau des pages – une page présentant un ou plusieurs contenus agencés - , ou encore au niveau d'un site complet. La gestion de version de site est complexe à manipuler, mais elle est intéressante pour publier conjointement des ensembles de contenus, répartis sur tout le site.

Qui parle de gestion des versions parle de gestion des conflits. Aussi, l'outil doit éviter toute perte, en interdisant l'accès à deux personnes simultanément pour le même document, ou en créant deux versions distinctes, une par personne.

Dans un souci de traçabilité, les contributeurs ont besoin d'une part de savoir qui a modifié quoi sur le site, et aussi de connaître les différences entres deux versions, que ce soit au niveau d'un contenu, d'une page, voire d'un site complet.

Un point différentiant entre les outils est la capacité à gérer les versions des fichiers joints, comme les fichiers PDF, mais aussi les images. Car quand un contributeur modifie un communiqué de presse avec un fichier Pdf joint, en remplaçant le texte mais aussi le Pdf, il faut que l'outil différentie les deux versions du Pdf, pour le circuit de validation et pour un possible retour en arrière.

Joomla et **Drupal** n'intègrent pas de système de gestion de versions. Joomla verrouille cependant les articles en cours d'édition afin d'éviter les conflits.

Spip conserve les versions des articles, et sait afficher les différences entre deux versions. Spip verrouille les articles lorsqu'un utilisateur les modifie, ce qui évite tout conflit. Le point faible de Spip est qu'il ne permet pas de modifier un article indépendamment de la version en ligne.

Typo3 est le seul CMS à gérer nativement des versions d'arborescences complètes. Depuis la version 3.7, un module de versionning permet en effet de créer des nouvelles versions de pages, mais aussi d'arborescences complètes, de les travailler, puis de les basculer en ligne. Notons que c'est sur ce module que s'appuie le workflow développé par Smile et publié en open source. Le différentiel reste au niveau de chaque contenu.

OpenCms gère les versions des contenus, et la séparation des espaces de travail et de publication permet de travailler une arborescence ou un contenu ou un fichier sans impacter la version en ligne.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

eZ publish sait gérer les versions de contenus, y compris les versions des fichiers associés aux contenus. eZ publish associe une version à un état dans le circuit de validation, ce qui permet de maîtriser parfaitement ses contenus, en modification et en suivi.

Jahia dispose d'un module de suivi des versions. La modification d'une page ou d'un de ses contenus, puis sa validation se font en parallèle de la version en ligne, avec même une prévisualisation de l'ensemble du site, dans sa version de travail.

InfoGlue gère les versions au niveau des contenus, et permet d'avoir pour un même contenu un état publié et un état brouillon.

Lenya gère deux arborescences de contenus. Une arborescence 'live' et une arborescence 'authoring', qui permettent de travailler un ensemble de contenus, de les prévisualiser et de les basculer en ligne en suivant le workflow. Un suivi détaillé des versions et de leurs modifications est intégré.

www.smile.fr

Cycle de vie des articles

Au sein de la base des contenus, les articles naissent, vivent et meurent parfois. Voyons comment les outils CMS gèrent ce cycle de vie.

L'un des besoins les plus classiques consiste à préparer un article *avant* sa date de publication, et de programmer sa mise en ligne automatique à une date donnée.

Un article peut également avoir une durée de vie connue à l'avance: soit en terme de durée, soit selon une date limite. On peut ainsi créer un article intitulé *"comment remplir votre déclaration d'impôts ?"*, et indiquer dès sa création que cet article est valable un an, ou bien jusqu'au 31 décembre 2008.

Que se passe-t-il alors à cette date ? L'article peut simplement être retiré de la publication de manière automatique. Dans certains cas, le contributeur souhaiterait être prévenu à l'avance de la péremption prochaine. Le CMS pourrait donc, dès la création de l'article, permettre de définir sa date de péremption, mais aussi si le contributeur souhaite être prévenu, combien de jours à l'avance, et si le webmaster également doit être prévenu. Une fois averti, le contributeur peut mettre à jour l'article si besoin, et définir une nouvelle date de péremption.

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Ce sont là des cycles de vie relativement simples. On peut souhaiter des choses plus sophistiquées, et en particulier qu'entre sa naissance et sa mort, les modalités d'apparition d'un article changent à certaines échéances. Il peut ainsi rester une semaine sur la home page, puis trois mois dans la rubrique actualités, puis un an dans la rubrique archives, et finalement disparaître.

Par rapport à cette gestion de cycles de vie, **Joomla** permet uniquement la définition de dates de début et de fin de vie des articles.

Drupal dispose d'un module « Scheduler » permettant de gérer le cycle de vie des contenus.

Spip répond au besoin standard et permet de publier des articles, préparés à l'avance, à partir d'une date donnée, mais sans date de fin de publication. Une modification légère permet de mettre en place une date de début et une date de fin de publication.

De la même manière, **Typo3** définit optionnellement une date de début et une date de fin de visibilité pour chaque article, sans alerte de fin de vie. Typo3 gère aussi la durée de vie sur une page, et même sur une arborescence complète.

eZ publish n'associe pas la durée de vie à un contenu, mais cette fonctionnalité est rendue possible par simple configuration de l'outil : il faut ajouter les 2 dates (ou plus) à chaque structure de contenu, et configurer l'événement de workflow pour qu'il compare la date du jour avec la date de début et la date de fin saisies.

Jahia permet de définir une date de début et une date de fin pour n'importe quel contenu. Pour des opérations plus sophistiquées, il est nécessaire de mettre en œuvre et de paramétrer le moteur de workflow.

Dans les métadonnées d'un contenu, **InfoGlue** propose une date de début et de fin, ce qui permet de définir un cycle de vie pour chacun des contenus.

Lenya ne propose pas de dates de début et de fin à un contenu, mais il est possible de les intégrer au contenu Xml, soit en métadonnées, c'est-à-dire saisies lors de la création, soit dans le contenu lui-même. Le workflow est à configurer pour qu'il tienne ensuite compte de ces dates.

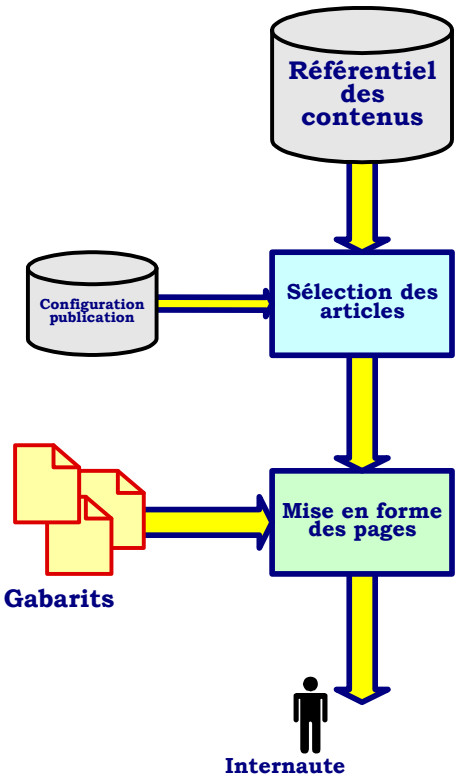
www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Publication

La restitution du contenu (*content delivery*) consiste à produire des pages sur un média donné, en intégrant le contenu au sein d'une mise en forme spécifique.

Le processus de publication peut être décomposé en deux étapes : sélection des contenus et mise en forme des pages, que nous décrivons successivement ci-dessous.



Sélection des contenus

Il est important de comprendre l'importance de l'étape de sélection des contenus.

Dans un site simple, on pourrait concevoir de ne restituer que deux types de pages : des pages de sommaire et des pages d'articles. Chaque page de sommaire ferait apparaître *la totalité* des sous-sections, et *la totalité* des articles associés à la section. Un lien sur chaque article donnerait accès à la page article.

Dans beaucoup de cas, une telle approche peut convenir. Mais il existe de nombreux besoins qui n'entrent pas dans ce schéma. Le plus classique est la restitution des *dernières actualités* en page d'accueil. D'une part la page d'accueil n'est pas la page sommaire de la rubrique *actualités* d'autre part on ne veut pas *toutes* les actualités.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

De même, dans un site de média, les journalistes contributeurs écrivent des articles qui viennent alimenter la base des contenus *au fil de l'eau*. Contrairement à la presse papier, il n'est pas possible de réagencer à la main les pages du site pour chaque nouvel article, en décidant de son positionnement. Il doit donc être possible de définir des modalités générales de sélection d'articles : *les trois derniers en date, ceux qui ont moins d'un an, les articles rédigés par tel contributeur, etc…* Ainsi, par rapport à la sélection d'articles, l'appartenance à une section est le critère par défaut, mais ce n'est pas le seul critère.

On note que ces possibilités enrichies de sélection d'articles répondent à d'autres problématiques de la publication. Par exemple elles peuvent contribuer à mettre en place un *cycle de vie spécifique*. Ou bien des *classifications transverses* i.e. qui ne dépendent pas de l'organisation principale du site.

Joomla permet de façon très souple et entièrement graphique de choisir les critères de sélection des contenu dans chaque partie du site : nombre, tri, pagination, tout est réglable sans aucun développement.

Drupal permet nativement de sélectionner le contenu par catégorie. Les gabarits PHP permettent de réaliser des sélections en utilisant des critères plus poussés.

Spip possède un système de squelettes (gabarits) spécifique, avec des balises intégrées au code Html. Assez puissant**,** il permet de définir de nouveaux tags simplement, et permet différentes opérations telles que les boucles imbriquées ou les affichages conditionnels.

Typo3 sépare moins les notions de base des contenus et de publication : les entités manipulées dans le back-office ne sont pas des articles, mais directement des pages, c'est à dire en référence directe à la restitution sur le site. En revanche, le mécanisme de gabarits de **Typo3** est particulièrement puissant : le gabarit Html proprement dit est complété par de la programmation en TypoScript qui permet de traiter les besoins les plus avancés, y compris en termes de sélection des contenus.

www.smile.fr

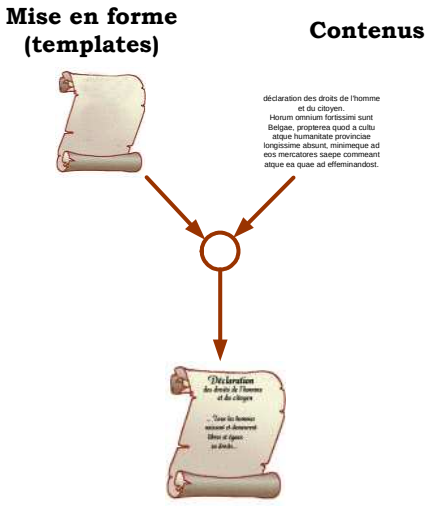
© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Les gabarits de **eZ publish** définissent la mise en page et la sélection des contenus. Des balises spécifiques permettent d'inclure des listes filtrées sur les éléments que l'on souhaite, pour afficher un menu ou autre. La définition de nouveaux tags se fait directement dans les gabarits, en langage php et en utilisant des fonctions natives de eZ publish. Chaque type de contenu a par ailleurs un gabarit spécifique, qui sera inséré au sein du gabarit global de la page.

La mise en page de **Jahia** est réalisée avec des gabarits JSP, entièrement paramétrables, que ce soit par bloc de fonctionnalités, ou directement dans le code JSP via les taglibs disponibles. L'agencement des blocs au sein de la page est réalisée en ligne, en mode administration, et l'on peut simplement organiser les contenus de la page.

www.smile.fr

Restitution et gabarits



La mise en forme des pages d'un site, pour une restitution Html, est toujours basée sur un dispositif de gabarits, ou *templates*, qui définissent la manière dont les contenus doivent être insérés dans des pages, et quelles indications de mise en forme accompagneront ces contenus.

Pour une page donnée, un ou plusieurs gabarits seront utilisés pour définir à la fois *quels contenus* sont sélectionnés, *où* ils sont positionnés dans la page, et *quels attributs de mise en forme* y sont associés. On pourra par exemple définir que telle page ne présentera que les champs *titre* et *accroche* d'une liste d'articles, que ces items seront insérés au sein d'un tableau placé sur la gauche de la page, et enfin que le titre sera en gras tandis que l'accroche sera en italique.

On voit bien sur ce petit exemple, la force de la séparation du contenu et de la mise en forme : il sera toujours temps de changer d'avis et de réagencer la page pour placer ces items à droite, et en couleur bleue, ou de définir une autre page, où chaque item est accompagné également d'une image.

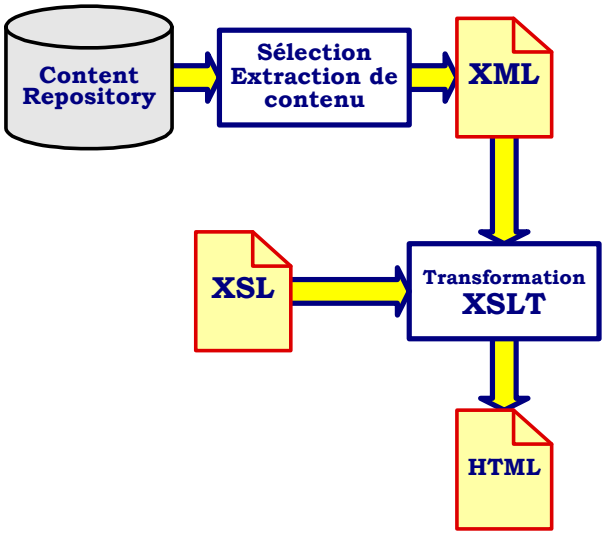
Il existe plusieurs techniques pour mettre en œuvre des gabarits.

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

La plus simple consiste à insérer dans un fond de page Html des marqueurs particuliers, des balises (en anglais *tags*), qui ne sont pas du Html, mais sont adressées au CMS. Le dispositif d'élaboration de pages du CMS va parcourir le gabarit, analyser ces balises, en déduire les contenus à sélectionner et à insérer à la position de la balise, réaliser l'insertion, puis envoyer la page. Par rapport à ce type de gabarits, il n'existe pas de standard quant à la syntaxe de ces balises, et chaque CMS définira les siennes.

L'avantage de ces premiers gabarits, que l'on appellera gabarits Html, est qu'ils sont pour l'essentiel constitués de Html, et donc assez faciles à manipuler par un webmaster ou un graphiste.

Les pages JSP et PHP sont également une forme de gabarits. Dans le cas des pages JSP, les insertions de contenu peuvent être définies soit par du code Java, soit par des tags faisant référence à des objets Java. Dans ce dernier cas, on parle de librairies de tags, ou *taglibs*. Le grand avantage des gabarits JSP est leur très large adoption pour le développement web Java en général, indépendamment de la problématique CMS. Dans le cas des pages PHP, le contenu est obligatoirement inséré par du code PHP. Il n'existe pas de système de *taglibs* dans ce langage.



La troisième voie est celle des gabarits Xsl, à base de Xml. Dans cette technique, le contenu est d'abord extrait et mis sous la forme d'un objet Xml, qui décrit chaque item de contenu à présenter dans la page. La mise en forme est définie au moyen d'un autre objet Xml, qui fait référence aux items de contenu du premier, et définit la manière dont ces items seront mêlés à des bribes de Html pour constituer la page. On utilise une *transformation Xsl*, qui est un outil standard, pour réaliser l'insertion du contenu, et produire les pages.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

La technique des feuilles de style Xsl et des transformations XslT présente des avantages importants :

- Elle s'appuie sur des standards reconnus, qui s'imposent au delà de la seule problématique de gestion de contenus.
- Elle offre un mécanisme puissant et très modulaire pour constituer les pages.
- Elle ne se limite pas au média Web, et permet aussi d'élaborer des pages destinées à d'autres canaux, en particulier l'élaboration de pages Pdf avec l'outil Xml-FOP, qui tend à devenir un standard.

En revanche, cette technique est plus complexe que celle des gabarits Html : le passage de la maquette Html au gabarit Xsl correspondant demande une certaine expertise, et ne pourra pas être confié à un graphiste ou un webmaster. On peut ajouter aussi que l'opération de transformation est relativement gourmande en CPU, et donc lente, mais la plupart des dispositifs de restitution intègrent un mécanisme de cache, de sorte que la transformation ne sera pas exécutée chaque fois qu'une page est demandée.

Joomla et **Drupal** utilisent des pages PHP qui permettent de créer des gabarits très complets, en utilisant toutes les fonctionnalités de ces CMS.

Spip utilise aussi des gabarits Html, avec ses balises spécifiques, mais qui peuvent être complétés le cas échéant par de la programmation PHP.

Typo3 possède également un système de gabarit Html, mais complété de manière intéressante par une configuration interactive, autorisant une certaine latitude de changements sur une simple interface d'administration : changements de couleurs, de fonte, et même d'agencement, peuvent être opérés sans toucher à un gabarit, au moyen d'un simple formulaire Html proposé au back-office.

OpenCms définit un système de gabarits à base de Xml, mais qui n'utilise pas des feuilles de style Xsl. La technique est puissante et bien structurée, mais elle présente l'inconvénient de n'être pas standard, et donc de ne pouvoir bénéficier des outils génériques qui apparaissent autour de ces standards.

eZ publish utilise des gabarits au format tpl, qui contiennent du code Html et des balises spécifiques à eZ publish. Les tags ont une syntaxe Php, et du code Php peut être directement inclus pour ajouter des fonctionnalités externes.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Les gabarits de **Jahia** sont en JSP, ce qui apporte souplesse et performance. Il est à remarquer que ce sont les mêmes gabarits qui permettent la restitution sur le site et l'administration des contenus. Le même gabarit doit disposer des trois différents modes (visualisation, modification et prévisualisation) et de la fonctionnalité de workflow. Il est en général possible de simplement décliner un gabarit, parmi les gabarits disponibles dans l'outil.

InfoGlue propose de réaliser les gabarits soit en velocity, un langage de template du projet Apache, soit directement en JSP, ce qui donne plus de souplesse et de possible interaction avec d'autres programmes.

Les gabarits de **Lenya** sont à base de feuilles de style Xsl, permettant de publier les contenus au format Html, mais aussi au format PDF ou même RTF. Le Xsl apporte une certaine complexité de mise en œuvre, notamment due à l'éclatement des templates dans le produit, rendant le travail d'intégration plus lourd.

Ajoutons aussi que, indépendamment du dispositif de gabarits, il existe une technique qui relève strictement du Html pour mettre en œuvre une bonne séparation du contenant et du contenu : les styles CSS. Le principe est de n'indiquer dans le Html que la *nature d'un élément*, et de déporter dans des feuilles de styles toutes les informations de présentation. La page Html dit que « *J. Hunter* » est le *nom de l'auteur*, et la feuille de style indique que le *nom de l'auteur* doit s'écrire avec telle mise en forme. Cette pratique, qui est clairement à généraliser, est indépendante d'un outil CMS en particulier.

Personnalisation de la restitution

La majorité des sites Internet présentent le même contenu à tous leurs visiteurs. Il existe toutefois différents cas de figure où l'on souhaite différencier la restitution selon les visiteurs et présenter à chacun ce qu'il souhaite voir, ou ce qu'il est autorisé à voir, à partir d'une base de contenu unique, administrée de manière globale.

On peut distinguer une *personnalisation à la demande du visiteur*, et une personnalisation définie *par les habilitations*.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

La première est typiquement la déclinaison du principe de portail à la manière de myYahoo, par lequel chaque internaute spécifie les contenus qui l'intéressent, et peut même configurer l'agencement de la page qui lui présentera ces contenus. Ayant défini ses préférences, chaque internaute sera reconnu, soit par une identification explicite, soit de manière transparente au moyen d'un cookie. Ce type de personnalisation est en particulier mis en œuvre au sein de portails d'entreprise, qui sont évoqués au chapitre suivant.

La personnalisation par habilitations n'est généralement pas gérée au niveau individuel, mais en créant des groupes de visiteurs, définis par leur *profil*. Cette personnalisation suppose bien sûr une véritable identification des internautes, qui permettra de retrouver le profil de chacun d'entre eux.

C'est essentiel en particulier sur des sites Intranets, où l'accès à certaines informations devra être restreint.

Joomla propose une personnalisation par habilitations sur 3 niveaux d'accès : public (tout le monde), membre (utilisateurs disposant d'un compte), et spécial (utilisateurs disposant d'un droit d'édition).

Drupal ne gère nativement que 2 niveaux d'accès pour la restitution : public et membre. Des modules supplémentaires apportent une plus grande souplesse et permettent de définir des droits détaillés pour chaque contenu.

Spip ne prévoit pas de restitution personnalisée. Il arrive que cette limitation soit rédhibitoire.

Typo3 gère de manière très complète les habilitations en lecture sur le site et permet donc une restitution personnalisée. En fait, c'est le même dispositif qui s'applique aux contributions et aux visites sur le site. Chaque page du site peut être affectée à un groupe d'utilisateurs et dans ce cas ne sera visible que par les visiteurs appartenant à ce groupe.

eZ publish gère les droits d'accès en restitution et la gestion des comptes utilisateurs. Le forum et les abonnements sont des fonctions qui relèvent de la personnalisation et qui sont mises en œuvre nativement.

Jahia intègre parfaitement les notions de personnalisation de la restitution, notamment en fonction des droits de l'utilisateur connecté, qui va voir ou non certains modules et certains modes de ces modules. Là encore, le mode 'portail' joue pleinement son rôle.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

InfoGlue gère les droits de lecture sur les contenus et les pages. Notons que InfoGlue est nativement connecté à CAS, une solution open source de gestion des identités (et de Single Sign On).

Apache Lenya permet de filtrer les accès aux contenus en fonction du profil de l'internaute.

Echanges de contenus et syndication

Il est courant qu'un site ait à échanger des contenus avec d'autres sites, soit qu'il reprenne des informations de sites partenaires, soit au contraire qu'il mette à disposition une partie de son contenu.

Le moyen normalisé de ces échanges consiste à mettre à disposition les contenus non plus au sein de pages Html, mais sous une forme Xml, qui permettra au site destinataire d'identifier sans ambiguïté chaque champ d'information.

Du point de vue légal, la récupération, et surtout la republication, du contenu d'autres sites ne peut se faire qu'avec l'autorisation du site émetteur, qui est à négocier au cas par cas, ou bien par l'intermédiaire d'entreprises de syndication.

Du point de vue technique, la mise à disposition d'informations au format Xml n'est pas très difficile, le mécanisme de *gabarits* permet de générer du Xml aussi facilement que du Html.

En revanche, l'insertion de contenus externes requiert un mécanisme spécifique pour analyser le flux Xml et l'insérer dans la base des contenus.

Pour que différents sites puissent se comprendre, il faut qu'ils décrivent leurs contenus selon le même langage. Il existe pour cela un standard : RSS ou Rich Site Summary / Really Simple Syndication, un format Xml, dont l'encadré ci-dessus fournit un exemple.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

On notera que l'insertion de contenus externes au moyen d'une interface RSS est une des fonctionnalités de base des portails à base de portlet, tels que Jetspeed ou uPortal, ou encore NetVibes.

Spip est particulièrement efficace en matière de syndication, puisqu'il sait nativement non seulement mettre à disposition un flux RSS, mais aussi intégrer un ou plusieurs flux RSS externes : il suffit de lui fournir l'url du RSS. De plus, Spip peut proposer à un administrateur la validation lien par lien des flux importés.

Joomla et **Drupal** permettent non seulement d'activer facilement des flux RSS pour le site et de configurer leur comportement (authentification, nombre d'éléments, ...)., mais également d'agréger des flux externes et de les afficher directement dans les pages du site.

Dans l'environnement **Typo3**, l'import RSS n'est pas natif non plus, mais une extension disponible permet d'exporter et d'importer des flux de syndication.

eZ publish permet de publier des informations dans plusieurs formats, dont le format RSS, ce qui lui permet d'être syndiqué par d'autres sites. L'import de flux RSS est possible, et l'import passera, si on le souhaite, par un processus de workflow avant d'être publié. Notons que eZ publish importe les flux dans des objets de contenus de type *lien*, ce qui nous permet de leur appliquer toutes les méthodes disponibles (multipositionnement, traduction, workflow, ...), et surtout de mélanger des flux de plusieurs émetteurs.

Jahia dispose d'un module d'import RSS, importés séparément dans la base de contenus. L'export RSS se fait par le moyen de gabarits à réaliser, dans lesquels il faut préciser les contenus à exporter.

InfoGlue ne propose pas nativement d'import ou d'export RSS. L'import et l'export Xml des contenus est par contre natif, qu'il faudrait adapter pour le réduire au format RSS.

Apache Lenya sait exporter les créations ou modifications de contenus au format RSS. L'import RSS est quant à lui mis en œuvre en utilisant une transformation XslT, permettant d'afficher en Html les liens du flux. En réalité, le flux n'est pas importé, il est juste affiché.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Mesure et suivi d'audience

Tout administrateur de site se doit d'être intéressé par l'audience, tant au global que détaillée pour chaque page. Il est possible que certaines pages ne soient pratiquement jamais vues, soit qu'elles n'intéresse personne, soit qu'elles soient mal mises en valeur, mal référencées, etc.

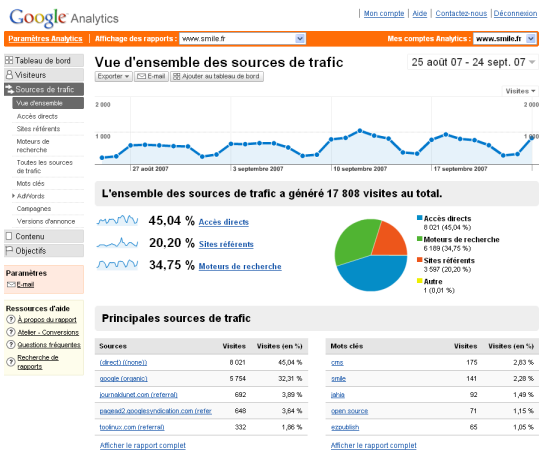
Suivre l'audience est donc absolument fondamental.

Il existe trois techniques pour le suivi de l'audience :

- L'analyse des logs du serveur web
- Un suivi externalisé, reposant sur des balises intégrées aux pages du site
- Une mesure intégrée au CMS lui-même.

L'analyse des logs du serveur est la technique la plus classique. Le serveur Http (Apache, IIS) écrit dans un fichier de log toutes les URL qu'il voit passer. Il existe de nombreux outils qui savent analyser ce fichier et en déduire une grande quantité d'information : nombre de pages vues, nombre de visites, jour par jour, heure par heure, d'où viennent les visiteurs, quel navigateur ils utilisent, quels pages sont les plus visitées, et bien d'autres choses encore.

Il existe de nombreux outils open source de qualité, tels que WebAlizer ou AWStats. A noter qu'en hébergement mutualisé, cette analyse est généralement fournie par l'hébergeur.



Le suivi externalisé est une technique plus récente, proposée par des prestataires spécialisés, tels que eStat ou Xiti, et maintenant Google Analytics, qui est gratuit et de bonne qualité. Elle consiste à placer dans les pages des balises spécifiques, invisibles au visiteur, et qui provoqueront des *hits* vers un serveur externe, spécialisé dans la mesure d'audience. L'administrateur peut se connecter au site du prestataire pour consulter de nombreux tableaux d'analyse de son audience. La forte concurrence entre les prestataires amène une grande richesse dans les états disponibles. Cette technique est donc puissante, mais d'une part elle présente un coût récurrent à payer au prestataire, et d'autre part elle implique le marquage des pages, qui peut être fait par l'intermédiaire des *gabarits*, mais requiert néanmoins un travail spécifique.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

A noter également l'outil PHPMyVisites qui permet de mettre en place un service semblable à Xiti sur ses propres serveurs.

Enfin quelques CMS relèvent eux-mêmes l'audience au moment où ils sont sollicités pour servir des pages. D'une manière générale, cette technique est beaucoup moins puissante que les deux précédentes : elle fournira beaucoup moins d'information sur les visiteurs et beaucoup moins d'analyses. Elle présente un seul avantage, c'est qu'elle permet simplement d'utiliser la connaissance de l'audience dans la restitution elle-même, par exemple en reprenant sur la page d'accueil les articles les plus consultés du moment.

Spip, Joomla, Drupal et **Typo3** relèvent eux-mêmes les hits sur les pages servies. Les statistiques (répartition des hits, mais également des informations sur les navigateurs ou les systèmes d'exploitation) sont disponibles directement dans le back office.

OpenCms, eZ publish, InfoGlue, Lenya et Jahia requièrent des dispositifs externes au CMS. Mais comme on l'a indiqué, ces solutions externes offrent des analyses sans comparaison, et sont donc recommandées dans tous les cas.

www.smile.fr

Indexation et moteur de recherche

On imagine trop souvent qu'un moteur de recherche se justifie principalement pour des indexations de documents en texte intégral, ou bien pour rechercher au travers de contenu peu organisé, par exemple dans les messages d'un forum.

C'est un usage restrictif. Beaucoup d'internautes utilisent la fonction de recherche simplement pour éviter de chercher à comprendre l'organisation du site. Ils préfèrent saisir les deux ou trois mots représentatifs de ce qu'ils cherchent et faire 'entrée', puis regarder ce qui leur est proposé, plutôt que de lire le premier menu, réfléchir pour trouver quelle est la rubrique la plus appropriée, parvenir à un niveau inférieur, réfléchir à nouveau, et finalement peut-être faire machine-arrière pour chercher ailleurs.

Ce comportement dans la recherche est inculqué aux internautes par les portails internet, qui sont le plus souvent le point de départ de leur navigation.

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Cette démarche directe, qui veut aller droit au but, sans traverser des pages intermédiaires et sans réfléchir, est parfois contraire au souhait du webmaster, qui aimerait que le visiteur prenne son temps, parcoure le plus de pages possibles, lise tout ce qu'on a préparé pour lui ici et là, et en outre, qu'il voie les publicités.

Mais ce que veut le webmaster ce n'est pas ce que veut le visiteur, et il faut se faire à l'idée que le visiteur aura le dernier mot puisque s'il n'est pas au but en trois clics, il ira voir ailleurs.

Un moteur de recherche est un outil spécifique qui ne relève pas à proprement parler du CMS. Les moteurs peuvent toujours être intégrés sur un site indépendamment de l'outil CMS retenu, dans la mesure où ils indexent le site par le http. Ils offrent tous des possibilités d'habillage des pages de résultat par un mécanisme semblable aux gabarits. Certains proposent également des API donnant accès directement aux fonctions de recherche, ce qui permet de mieux les intégrer au sein d'applications spécifiques.

L'intégration de moteurs de recherche externes sera difficile dans le cas où des accès restreints à certains contenus sont mis en place, car d'une part, le moteur doit indexer ces contenus protégés, et d'autre part, le moteur doit vérifier les habilitations de l'internaute pour afficher les résultats de recherche en y incluant ou pas les contenus protégés.

Dans le monde open source, le produit phare est bien sûr HtDig, qui est assez dynamique et très solide. Depuis peu, un autre produit lutte pour le leadership. Il s'agit de mNoGoSearch, un produit d'origine russe , avec un développement logiciel très actif, ce qui faisait défaut à htDig.

Citons également les produits Lucene et Nutch, qui sont des solutions tout à fait solides, en environnement Java.

Spip offre des fonctions de recherche intégrées mais très simplifiées. C'est une recherche très sommaire, qui ne traite pas les opérateurs logiques, l'indexation des documents, et le classement par pertinence. Pour un besoin sérieux, notre recommandation est de recourir à un moteur externe.

Joomla permet des recherches selon 3 modes : « AND », « OR », et « expression exacte ». Les résultats peuvent être triés suivant la date de publication (plus récent/plus ancien), la popularité , l'ordre alphabétique ou la section/catégorie.

Drupal offre une recherche avec opérateurs logique et permet de filtrer les résultats selon leur catégorie et leur type de contenu. Grâce au module « Search Atttachements », Drupal peut également effectuer des recherches full-texte dans les documents attachés.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

eZ publish offre une fonction de recherche plus avancée, avec notamment la possibilité de rechercher par champ de contenu (titre=…), mais aussi une recherche plein-texte sur les contenus et les documents joints. Seule la complétion de mots n'est pas intégrée.

Typo3 intègre un moteur de recherche qui indexe réellement les contenus, y compris les documents joints, et la liste des résultats est triée sur un critère de pertinence calculé. Ce module reste inférieur à htDig ou mNoGoSearch dans le sens où il ne permet pas de recherche phonétique ou approchante, mais il est très performant, et propose des résultats organisés par dossier. Pour des besoins avancés, un moteur de recherche externe restera nécessaire.

Jahia intègre le moteur Lucene issu du projet Apache, qui permet d'indexer l'ensemble des contenus du site, sans utiliser le http mais avec des connecteurs spécifiques pour la lecture des fichiers et de la base de données.

InfoGlue intègre le moteur de recherche Lucene, qui parcourt les pages du site et les indexe, proposant ainsi une recherche full-text sur l'ensemble des contenus. La recherche sur les contenus à accès protégés reste à mettre en oeuvre.

Lenya intègre Lucene, le moteur de recherche de la même fondation Apache, qui indexe les contenus en full-text, et permet des recherches en tenant compte des droits de lecture de l'internaute.

Il faut évoquer aussi la question de l'indexation par des moteurs de recherche externes, tels que Google.

Les conditions de base pour qu'un site puisse être indexé sont (1) que toutes les pages soient accessibles sans passer par un formulaire – les moteurs d'indexation ne franchissent jamais un formulaire, même simple, et (2) que les URL soient stables et en particulier qu'elles n'incluent pas de jeton de session comme le font certains outils de développement.

Par ailleurs, chaque page du site peut définir la balise *meta/keywords*, qui permet d'annoncer explicitement les mots clés caractérisant la page. Avec les dispositifs de gabarits, il est toujours possible de définir le contenu de cette balise à partir de l'un des champs décrivant l'article. Encore faut-il y penser.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Gestion des url

Les CMS affichent des contenus dynamiquement aux internautes. A ce titre, certains CMS produisent des urls parsemées de paramètres permettant de définir la page appelée. On aura par exemple une url du type www.monsite.fr/article.php3?id_article=18.

Certains CMS ont souhaité produire des urls compréhensibles, qui donnent une indication quant à la page affichée. Les urls obtenus sont alors du type www.monsite.fr/finances/resultats-financiers.htm pour le contenu 'résultat' de la rubrique 'finances'.

Le deuxième type d'url, porteur de sens, présente l'avantage d'être compréhensible par l'internaute qui sait qu'il est dans la rubrique 'finances' seulement en voyant l'url. Ce type d'url présente surtout l'avantage d'être interprété par les moteurs d'indexation tels que Google, qui va donner plus ou moins de points à cette page en fonction des mots présents dans l'url. On parlera d'urls 'propres' ou 'significatives' ou encore d'urls 'Search Engine Friendly'.

Le premier type d'url a l'avantage d'être stable. Si l'article 18 est déplacé dans la rubrique 'finances' à 'résultats', son url restera la même. Alors qu'avec le deuxième type, l'url de l'article sera changée, ce qui peut provoquer des liens cassés dans certains cas.

Tous les CMS s'orientent aujourd'hui vers le deuxième type d'url, même si certains vont plus loin que d'autres.

Deux points nous semblent importants dans la mise en œuvre des urls propres : Tout d'abord, la possibilité pour le contributeur de personnaliser l'url d'un contenu de manière à maîtriser encore plus l'indexation par les moteurs tels que Google. Et ensuite, la gestion par le CMS de la stabilité de l'url. Le CMS doit éviter les liens cassés, en stockant l'historique des urls d'un contenu, et en les maintenant opérationnelles même après que le contenu ait été modifié.

Spip permet depuis peu d'obtenir des urls propres. Les urls obtenues ne simulent pas l'arborescence des contenus, mais permettent tout de même une meilleure indexation.

Joomla et **Drupal** permettent de mettre en œuvre les url propres pour l'intégralité du site par une simple configuration dans l'interface d' administration.

eZ publish va simuler dans l'url toute l'arborescence du contenu, et va permettre de modifier pour chaque contenu son url. De plus, eZ publish va maintenir opérationnelles les urls passées d'un contenu, évitant tout lien cassé.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

Typo3 permet de réaliser des urls propres, grâce à un module à implémenter, développé par l'équipe centrale. Le module permet de plus d'avoir une url propre par langue de contenu.

Jahia permet de définir pour chaque page un alias, permettant de créer des urls propres simulant une arborescence. La mise en œuvre se fait cependant page à page.

InfoGlue définit les urls propres pour les pages, qui sont des regroupements de contenus. Dans ce sens, les urls

Lenya permet nativement d'obtenir des urls propres, mais ne va pas gérer l'historique des versions des urls.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation

SYNTHESE

Avant de conclure, soulignons que notre analyse était construite autour des *concepts fondamentaux* de la gestion de contenu, et non autour des *critères de choix.* Notre objectif n'était pas ici de construire un tableau croisé évaluant chacun des CMS du panel sur chacun des critères. Notre objectif, comme nous l'avons indiqué en préambule, était de proposer notre vision de la problématique de gestion de contenu.

Les critères de choix, quant à eux, se réfèrent pour partie à ces concepts fondamentaux, mais incluent aussi d'autres facteurs : dynamisme de la communauté, références, performances, documentation, etc.

Il n'y a pas un outil qui ressortirait comme le gagnant de l'étude : la sélection s'est faite en amont, c'est à dire que nous n'avons présenté ici que des outils de qualité, qui sont parmi les meilleurs choix à ce jour pour construire votre site, ou bien gérer plus largement vos contenus.

Les solutions open source sont à l'informatique ce que les médicaments génériques sont à la pharmacie : une fois qu'un domaine est devenu mature, que les concepts sont stabilisés, les bonnes pratiques et les solutions relèvent du domaine public. C'est progressivement ce qui est en train de se passer dans le domaine de la gestion de contenu. Sur le bas de gamme, la cause est entendue et il n'y a pratiquement plus d'offre commerciale. Sur le haut de gamme, certains produits offrent une alternative tout à fait crédible aux offres commerciales.

Smile a mis en œuvre des solutions de gestion de contenu sur plusieurs centaines centaine de sites, en Internet ou intranet, et peut mettre son expertise au service de vos projets.

www.smile.fr

© Copyright Smile - Open Source Solutions - Toute reproduction interdite sans autorisation